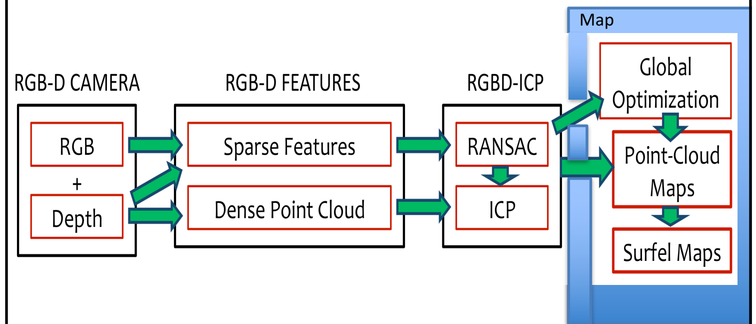
CSE 571
Robotics

RGB-D Mapping

University of Washington
Dieter Fox

RGB-D Mapping Overview



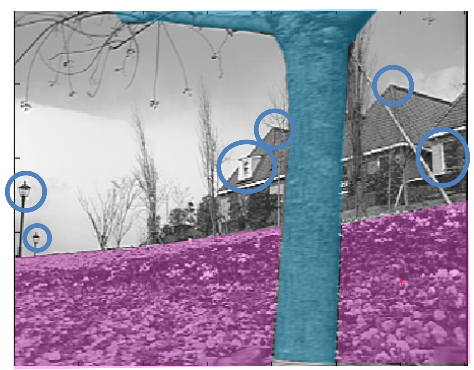
RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. Henry et al. ISER 2010
RGB-D Mapping: Using Kinect-style Depth Cameras for Dense 3D Modeling of Indoor Environments. Henry et al. IJRR 2012

Visual Features

- Detector
  - Repeatable
  - Stable
  - Invariances:
    - Illumination
    - Rotation
    - Scale
- Descriptor
  - Discriminative
  - Invariant

Visual Features

- Tree bark itself not really distinct
- Rocky ground not distinct
- Rooftops, windows, lamp post fairly distinct and should be easier to match across images



Say we have 2 images of this scene we'd like to align by matching local features

What would be good local features (ones easy to match)?

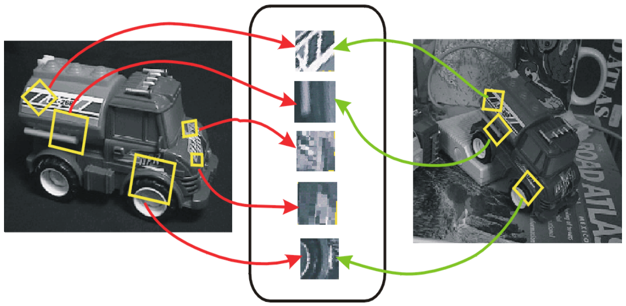
Courtesy: S. Seitz and R. Szeliski

Invariant local features

-Algorithm for finding points and representing their patches should produce similar results even when conditions vary

-Buzzword is "invariance"

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …



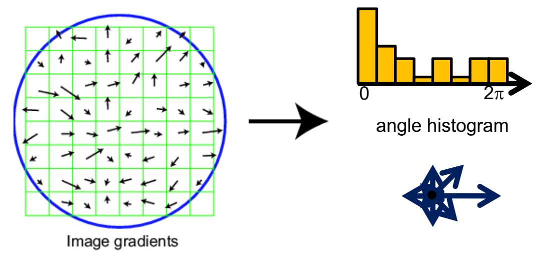
Feature Descriptors

Courtesy: S. Seitz and R. Szeliski

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute gradient for each pixel
- Throw out weak gradient magnitudes
- Create histogram of surviving gradient orientations

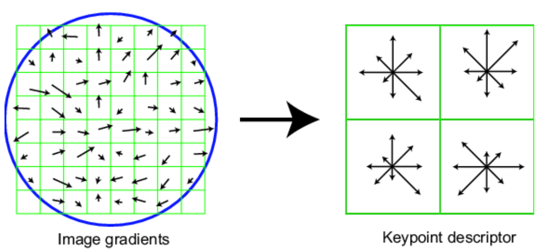


Adapted from slide by David Lowe

SIFT keypoint descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Adapted from slide by David Lowe

Properties of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - http://www.vlfeat.org
  - http://www.cs.unc.edu/~ccwu/siftgpu/

Feature distance

- How to define the difference between two features f1, f2?
  - Simple approach is SSD(f1, f2)
    - sum of square differences between entries of the two descriptors
    - can give good scores to very ambiguous (bad) matches

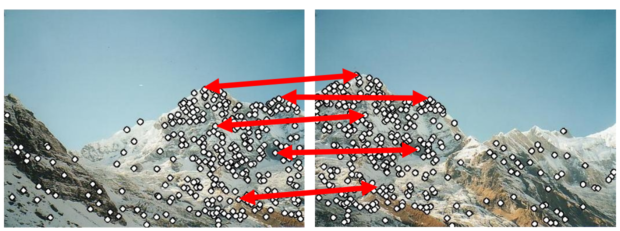
I_1 I_2

Feature distance

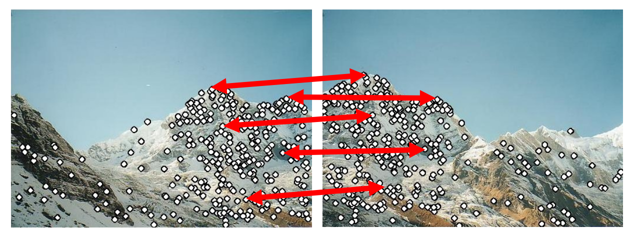
- How to define the difference between two features f1, f2?
  - Better approach: ratio distance = SSD(f1, f2) / SSD(f1, f2')
    - f2 is best SSD match to f1 in I2
    - f2' is 2nd best SSD match to f1 in I2
    - gives small values for ambiguous matches

I_1 I_2

Are descriptors unique?



Are descriptors unique?



No, they can be matched to wrong features, generating outliers.

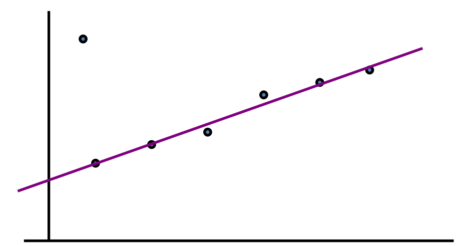
Strategy: RANSAC

- RANSAC loop:
  1. Randomly select a *seed group* of matches
  2. Compute transformation from seed group
  3. Find *inliers* to this transformation
  4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers
- Keep the transformation with the largest number of inliers

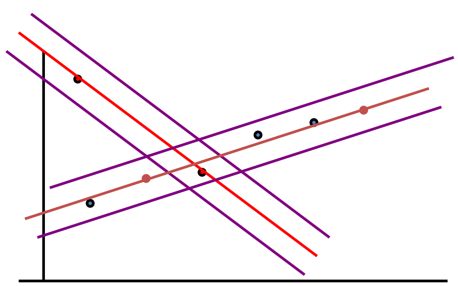
M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

Simple Example

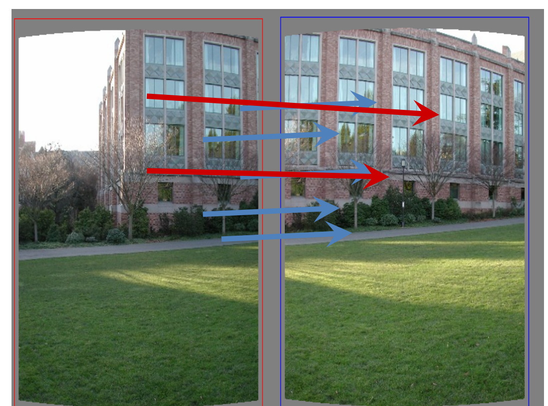
- Fitting a straight line



Why will this work ?



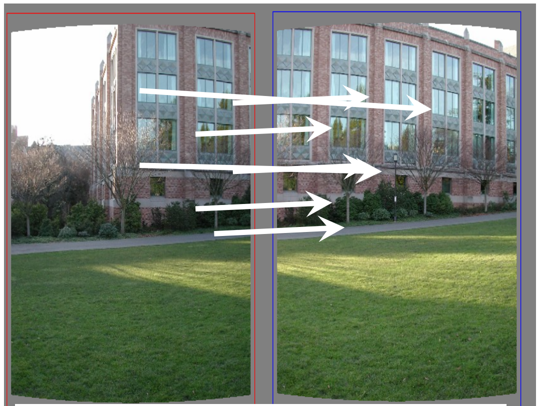
RANSAC example: Translation



Putative matches

Slide: A. Efros

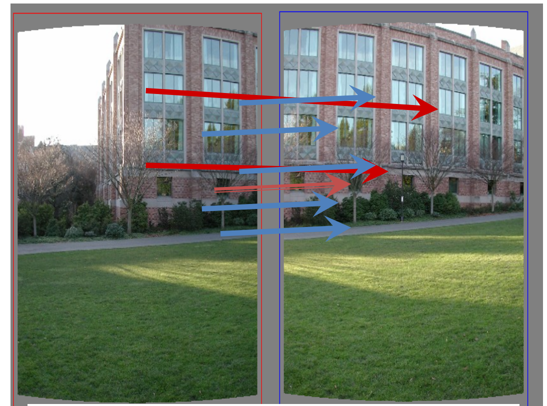
RANSAC example: Translation



Select *one* match, count *inliers*

Slide: A. Efros

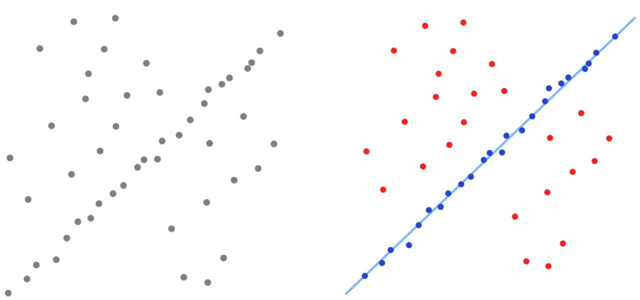
RANSAC example: Translation



Find “average” translation vector

Slide: A. Efros

RANSAC: Line Fitting

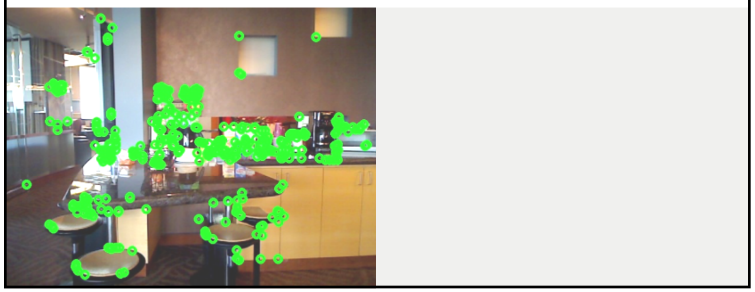


RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to tune
  - Can’t always get a good initialization of the model based on the minimum number of samples
  - Sometimes too many iterations are required
  - Can fail for extremely low inlier ratios

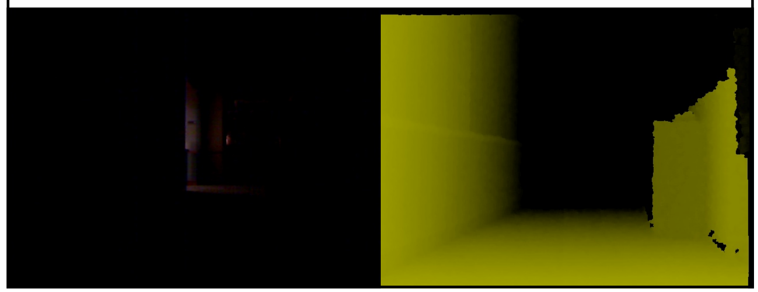
Visual Odometry

- Compute the motion between consecutive camera frames from visual feature correspondences.
- Visual features from RGB image have a 3D counterpart from depth image.
- Three 3D-3D correspondences constrain the motion.



Visual Odometry Failure Cases

- Low light, lack of visual texture or features
- Poor distribution of features across image
- But: RGB-D camera still provides shape info



ICP
(Iterative Closest Point)

- *Iterative Closest Point* (ICP) uses shape to align frames
- Does *not* require the RGB image
- Does need a good initial "guess"
- Repeat the following two steps:
  - For each point in cloud A, find the closest corresponding point in cloud B
  - Compute the transformation that best aligns this set of corresponding pairs

ICP Variants

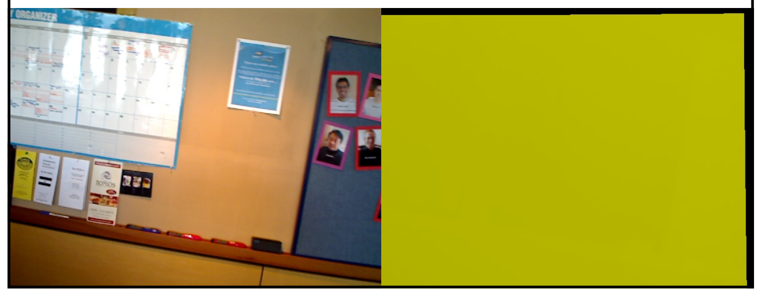
- Correspondence
  - Outliers as absolute or percentage
  - No many-to-one correspondences
  - Reject boundary points
  - Normal agreement
- Error metric
  - Point-to-point
  - Point-to-plane
  - Weight by color / normal agreement

ICP (Iterative Closest Point)

- Iteratively align frames based on shape
- Needs a good initial estimate of the pose

ICP Failure Cases

- Not enough distinctive shape
- Don't have a close enough initial "guess"
- Here the shape is basically a simple plane…



Optimal Transformation

- Jointly minimize feature re-projection and ICP:

$$\mathbf{t}^* = \operatorname*{argmin}_{\mathbf{t}} \left[\left(\frac{1}{|A_f|} \sum_{i \in A_f} \left| Proj(\mathbf{t}(f_s^i)) - Proj(f_t^i) \right|^2 \right) + \beta \left(\frac{1}{|A_d|} \sum_{j \in A_d} w_j \left| (\mathbf{t}(p_s^j) - p_t^j) \cdot n_t^j \right|^2 \right) \right]$$

RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. Henry et al. ISER 2010

RGB-D Mapping: Using Kinect-style Depth Cameras for Dense 3D Modeling of Indoor Environments. Henry et al. IJRR 2012

Joint Optimization (RGBD-ICP)

input : source RGB-D frame P_s, target RGB-D frame P_t, previous transformation $\mathbf{T}_p$
output: Optimized relative transformation $\mathbf{T}^*$

```
F_s ← Extract_RGB_Point_Features (P_s)
F_t ← Extract_RGB_Point_Features (P_t)
(T*, A_f) ← Perform_RANSAC_Alignment (F_s, F_t)
if |A_f| < γ then
    T* = T_p
    A_f = ∅
end
repeat
    A_d ← Compute_Closest_Points (T*, P_s, P_t)
    T* ← Optimize_Alignment (T*, A_f, A_d)
until (Change (T*) ≤ θ) or (Iterations > MaxIterations)
return T*
```

Algorithm 1: RGB-D ICP algorithm for matching two RGB-D frames.

Experiments

- Reprojection error is better for RANSAC:

	EE-RANSAC	RE-RANSAC
Mean inliers per frame	60.3	116.7

- Errors for variations of the algorithm:

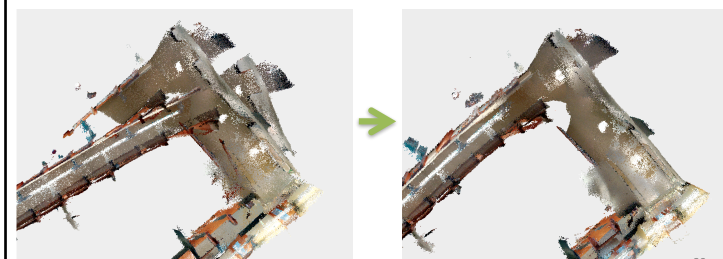
	RE-RANSAC	EE-RANSAC	ICP	RGBD-ICP	Two-Stage RGBD-ICP
Intel-Day	0.11 (±0.05)	0.16 (±0.07)	0.15 (±0.05)	**0.10** (±**0.04**)	0.11 (±0.05)
Intel-Night	1.09 (±0.88)	1.15 (±0.89)	0.17 (±0.06)	**0.15** (±**0.08**)	0.15 (±0.09)

- Timing for variations of the algorithm:

	RE-RANSAC	EE-RANSAC	ICP	RGBD-ICP	Two-Stage RGBD-ICP
Intel-Day	0.21 (±0.03)	0.20 (±0.05)	0.72 (±0.73)	0.48 (±0.10)	0.21 (±0.03)
Intel-Night	0.20 (±0.05)	0.20 (±0.05)	0.43 (±0.64)	0.57 (±0.47)	0.37 (±0.63)

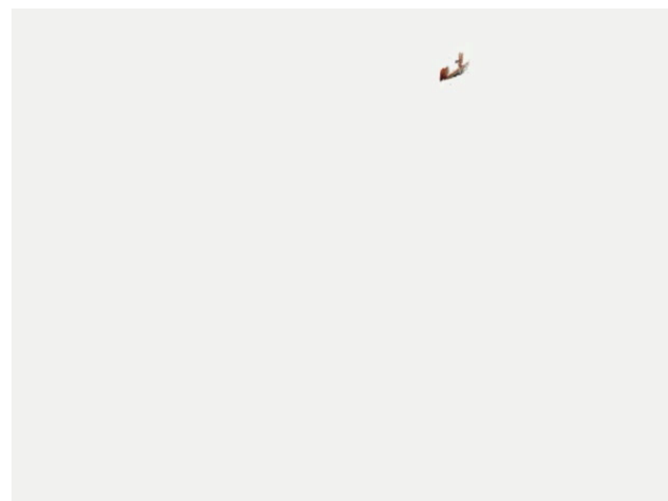
Loop Closure

- Sequential alignments accumulate error
- Revisiting a previous location results in an inconsistent map



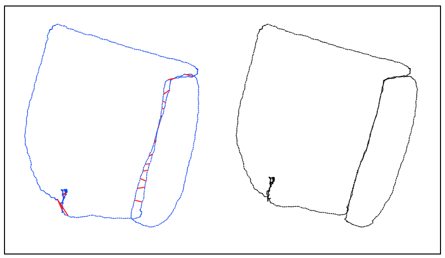
Loop Closure Detection

- Detect by running RANSAC against previous frames
- Pre-filter options (for efficiency):
  - Only a subset of frames (*keyframes*)
  - Only keyframes with similar estimated 3D pose
  - Place recognition using vocabulary tree
    - *Scalable recognition with a vocabulary tree*, David Nister and Henrik Stewenius, 2006
- Post-filter (avoid false positives)
  - Estimate maximum expected drift and reject detections changing pose too greatly



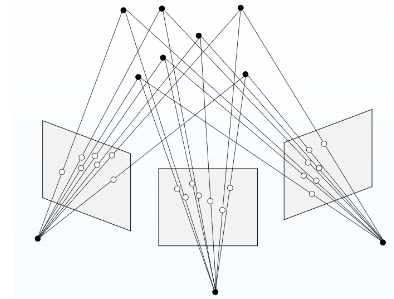
Loop Closure Correction (TORO)

- TORO [Grisetti 2007, 2009]:
  - Constraints between camera locations in *pose graph*
  - Maximum likelihood global camera poses



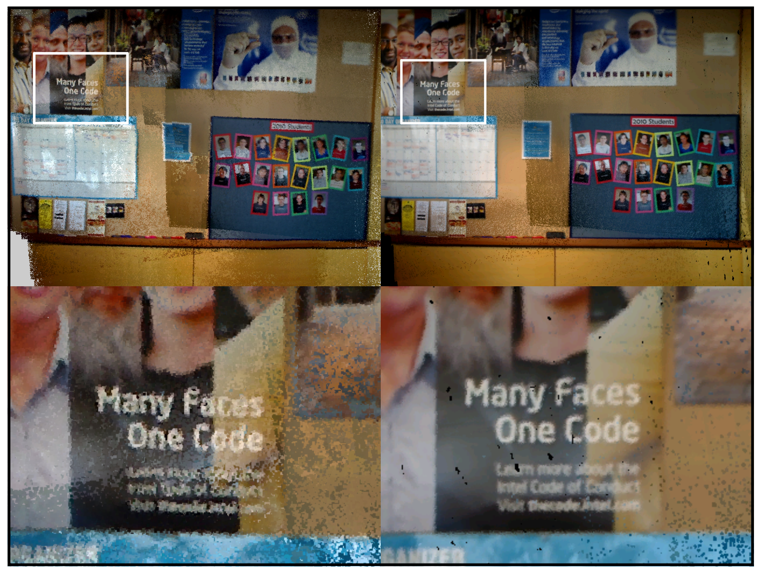
[Image: Manolis Lourakis]

Loop Closure Correction: Bundle Adjustment



$$\sum_{c_i \in C} \sum_{p_j \in P} v_{ij} |Proj(c_i, p_j) - (\bar{u}, \bar{v}, \bar{d})|^2$$

SBA Points



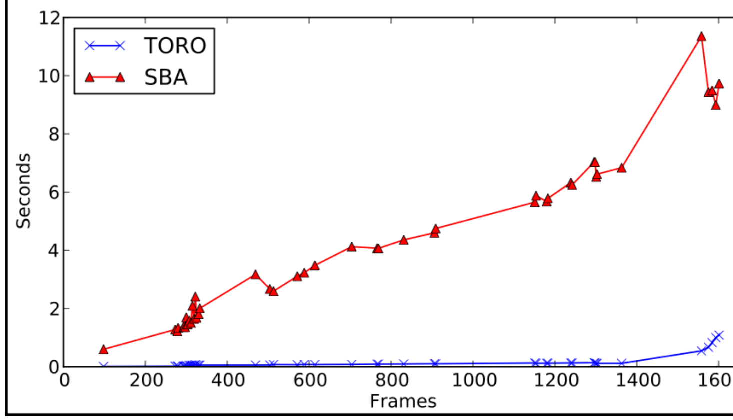
A Second Comparison

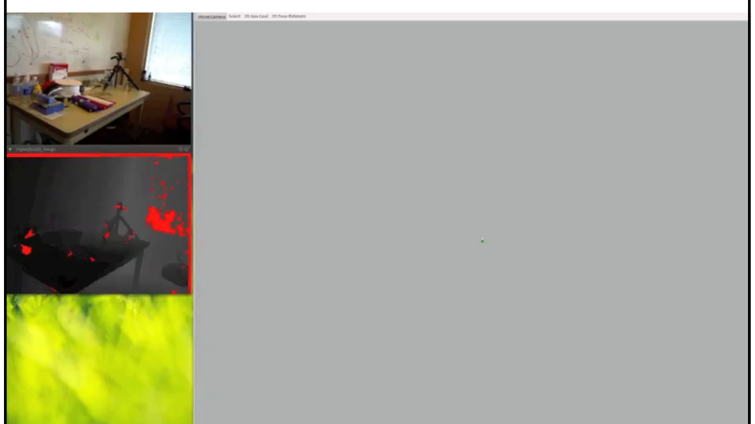
TORO

SBA

37

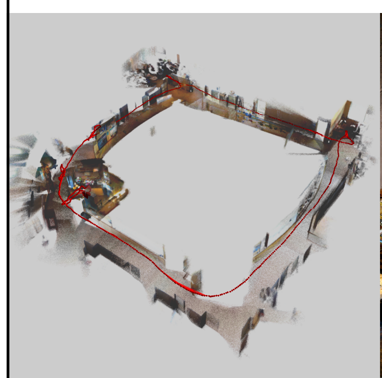
Timing

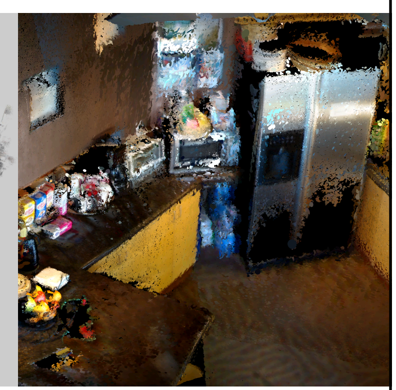




39

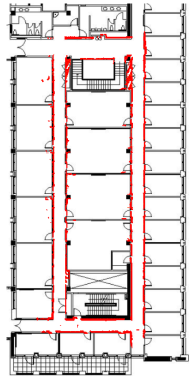
Resulting Map





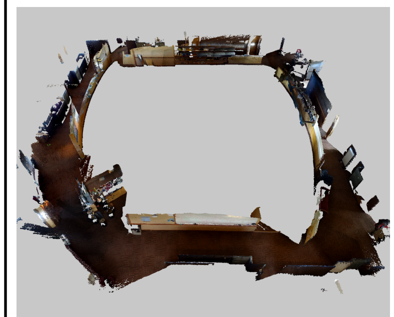
40

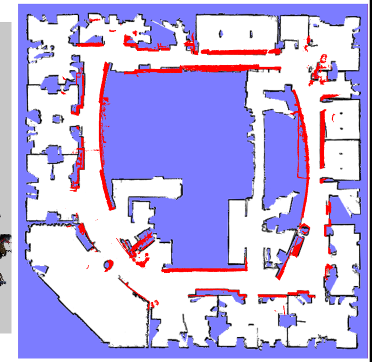
Experiments: Overlay 1



41

Experiments: Overlay 2





42

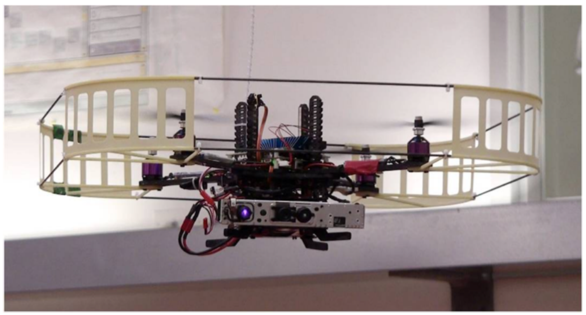
Map Representation: Surfels

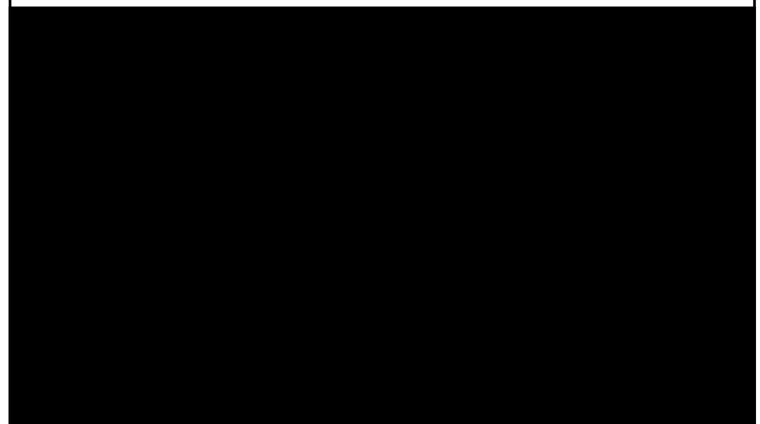
- *Surface Elements* [Pfister 2000, Weise 2009, Krainin 2010]
- Circular surface patches
- Accumulate color / orientation / size information
- Incremental, independent updates
- Incorporate occlusion reasoning
- 750 million points reduced to 9 million surfels

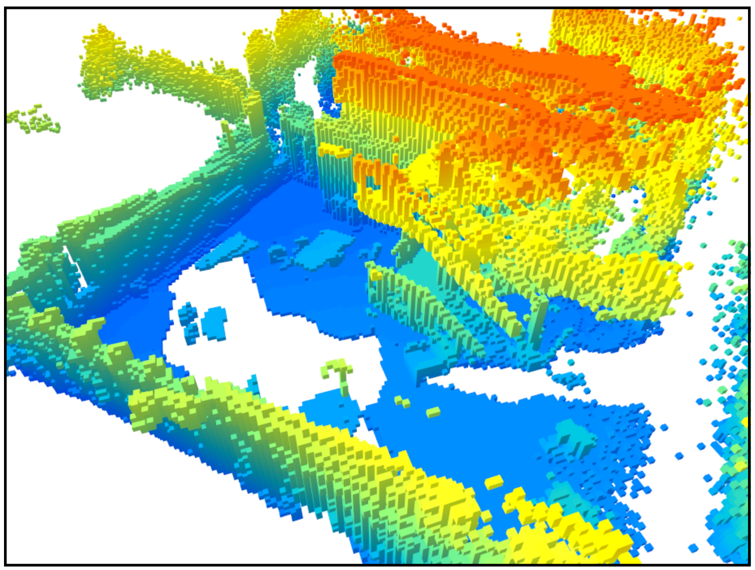
43

Application: Quadrocopter

- Collaboration with Albert Huang, Abe Bacharach, and Nicholas Roy from MIT







Larger Maps

[Whelan-Leutenegger-SalasMoreno-Glocker-Davison: RSS-15]

ElasticFusion

ElasticFusion: Dense SLAM Without A Pose Graph

Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, Andrew Davison

Imperial College London

50

Conclusion

- Kinect-style depth cameras have recently become available as consumer products
- RGB-D Mapping can generate rich 3D maps using these cameras
- RGBD-ICP combines visual and shape information for robust frame-to-frame alignment
- Global consistency achieved via loop closure detection and optimization (RANSAC, TORO, SBA)
- Surfels provide a compact map representation

51