

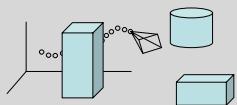
# Wide-Area 3D Localization from Known 3D Structure

Olivier Koch, Seth Teller  
Massachusetts Institute of Technology, MIT CSAIL

## Problem Statement

Determine the 6-DOF pose of an omnidirectional camera given a coarse 3D map of the structure.

- ✓ in 3D
- ✓ across multiple buildings
- ✓ under real conditions



## Novelty of the Method

- ✓ Solves for initialization.
- ✓ Uses offline computation of visible lines.
- ✓ Scales to large cluttered environments.
- ✓ Does not rely on Structure from motion.

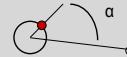
## References

- Linear pose estimate from points or lines*, Ansar & Daniilidis, ECCV '02.  
*Structure from motion using lines*, Bartoli & Sturm, CVIU '05.  
*Real-time visual tracking of complex structures*, Drummond & Cipolla, PAMI '02.  
*Visual Odometry*, D. Nister, CVPR '04.  
*Fusing points and lines for high performance tracking*, Rosten & Drummond, ICCV '05.  
*Robot Vision*, BKP Horn, '86.  
*Multiple View Geometry in Computer Vision*, Hartley & Zisserman, '04.

## Our Approach

Initialize and track a set of correspondences between 3D model lines and 2D image edges.

Score using the bearing  $\alpha$  between the 3D model line (○) and the 2D image edge (●).

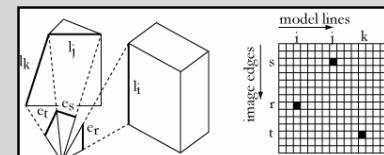


### The Initialization Phase

How to align a set of 2D edges onto a set of 3D model lines ?

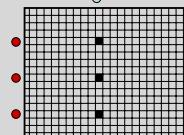


We propose a novel probabilistic approach based on the dihedral angle ("bearing") between pairs of model lines and pairs of image edges.



When three model lines and three image edges have the same bearing configuration, the matching table is updated.

For each model line, we extract the most probable edge matches.

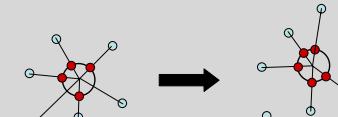


We generate random samples of line-edge correspondences and score them.

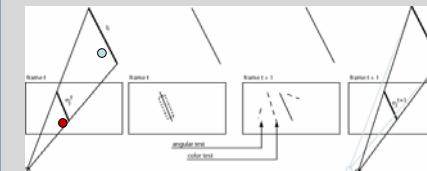
Sample 1 = { (○,●), (○,●), (○,●), (○,●), ..., (○,●) }  
⋮  
Sample n = { (○,●), (○,●), (○,●), (○,●), ..., (○,●) }

## The Maintenance Phase

How to update a set of correspondences from frame t to frame t+1 ?



We suggest a hue-based correspondence tracker coupled with a state machine filter.



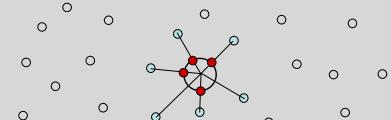
Each correspondence is updated on the image and in the state machine. The new camera pose is computed from the new set of correspondences.

A newcomer correspondence is set to Unknown and promoted to Accepted if it is consistently observed.

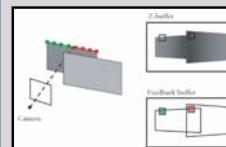


## Visibility Pre-computation using the GPU

How to bound the search space of model lines in the structure ?



We subdivide the structure into nodes and compute the set of lines visible from each node using an OpenGL algorithm.



Depth values agree in the Z-buffer and the feedback buffer when the 3D point is visible (green) and disagree due to occlusion otherwise (red).

## Results

Our system is demonstrated on several real datasets.



**Initialization**  
A model line and the edge matches guessed by the probabilistic algorithm.



A successful initialization from a single omnidirectional image. Structure reprojected in green.



**Maintenance**  
3D ego-motion on a long-run excursion through a cluttered lab (1,800 frames – 5 min).



3D ego-motion on a long-run excursion through several buildings (7,000 frames – 25 min).



A snapshot of the maintenance phase. Model lines are color-coded according to their state (green = accepted, orange = pending, red = unknown).



## An Application to 3D model texture painting

Pixels colors are back-projected onto the 3D model from the known camera position. Here: Stata 32-337.

## Future Work

- ✓ Signature-based initialization
- ✓ Integration of inertial sensor units
- ✓ Online update of the 3D model

## Acknowledgements

We are grateful to the Charles Stark Draper Laboratories for Internal Research and Development funding of this work.