Wide-Area Egomotion from Coarse 3D Structure and Omnidirectional Video

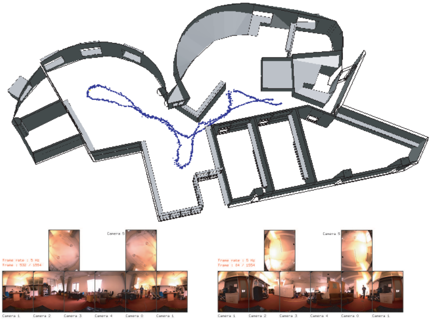
Olivier Koch
koch@csail.mit.edu

February 19, 2007

Problem Statement

Achieve robust, accurate 3D vision-based localization :

- over large environments (several buildings)
- under real, cluttered conditions
- in the 6-DOF space
- with close to real-time performance



Localization

Initialization
Global
Once at startup

Maintenance
Local
At every frame

+ light-weight “relock” mechanism when lock is lost

Target Capabilities

Robust:

- must work for 30 min without breaking
- "relock" successfully within a minute

Accurate:

- 2 degrees in rotation, a few inches in translation

Close to real-time:

- Initialization in a few minutes, Maintenance at 1Hz

Related Work

Theoretical background:

- BKP Horn, Robot Vision, MIT Press 1986.
- Faugeras et al., The Geometry of Multiple Images, MIT Press, 2001.
- Hartley and Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

Academic papers:

- Dhome et al., Determination of the Attitude of 3d objects from a single perspective view, PAMI 89.
- Quan and Dan, Linear n-point camera pose determination, PAMI 99.
- Taylor and Kriegman, Structure and Motion from line segments in multiple images, PAMI 95.

Related Work

Academic papers:

- Ansar & Daniilidis, Linear Pose Estimation from Points or Lines, EECV 02.
- Bartoli et al.,Motion from 3D line correspondences: linear and non-linear solutions, CVPR 03.
- Drummond and Cipolla, Real-time visual tracking of complex structures, PAMI 02.
- Nister, An efficient solution to the five-point relative pose problem, PAMI 04.
- Rosten and Drummond, Fusing points and lines for high performance tracking, ICCV 05.
- Gordon and Lowe, Scene modelling, recognition and tracking with invariant image features, ISMAR 04.

Contributions

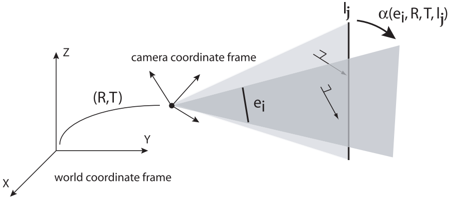
- solves for initialization;
- robust to significant clutter and transient motion.
- uses omnidirectional images for full view freedom;
- scales to large, real-world environments;
- does not drift.

Approach: Pose from Line Correspondences

Our method matches 2D image edges with 3D model lines:

- does not perform Structure from Motion (SFM);
- relatively unexplored (vs point features);
- robustly detectable;
- intuitively more precise than point features.

Approach: Pose from Line Correspondences



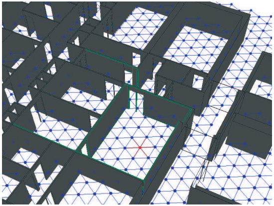
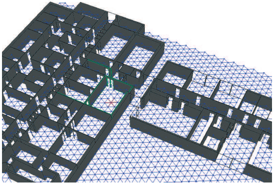
$$\xi(R, T) = \frac{1}{n} \cdot \sum_{i=1}^{n} \alpha(e_i, R, T, l_j)^2 \tag{1}$$

where R and T are the rotational and translation components of the camera's rigid-body pose expressed with respect to the model coordinate origin, n is the number of correspondences, and α is the angle between the two planes spanned by the camera center and the observed image edge e_i and model line l_j respectively.

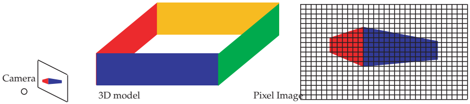
Offline Visibility Precomputation

We pre-compute the set of visible faces and lines in the model.

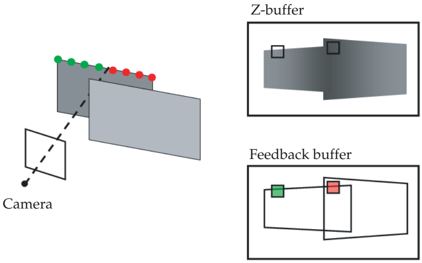
- drastically speeds up the localization process
- minimizes the risk of mismatches
- uses a model grid subdivision

- Determine the set of visible faces using an OpenGL renderer:



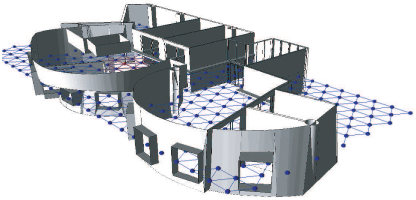
- Determine the set of visible lines using OpenGL framebuffer:



Initialization

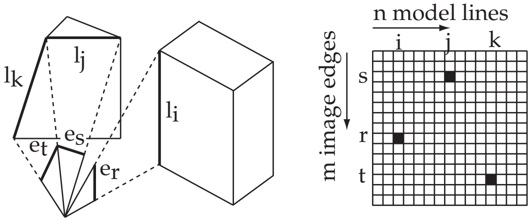
From a single omnidirectional image:

- requires a search region estimate from the user
- uses a single omnidirectional image
- find the most probable camera pose (minimize ξ).
- initialize a set of correspondences



Probabilistic Edge-Line Matching

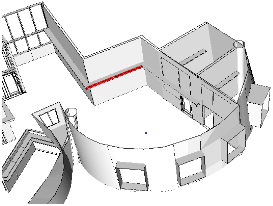
1. We test whether a triplet of 2D edges can be a possible match for a triplet of 3D lines given a camera search region.



- We process all triplets of edges/lines and populate a correspondence probability matrix.

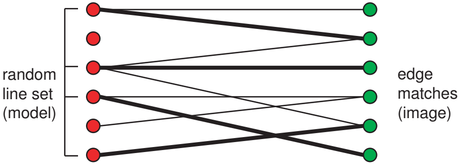
Probabilistic Edge-Line Matching

2. For each model line, we keep the top-k candidate edges ($k \sim 4$).

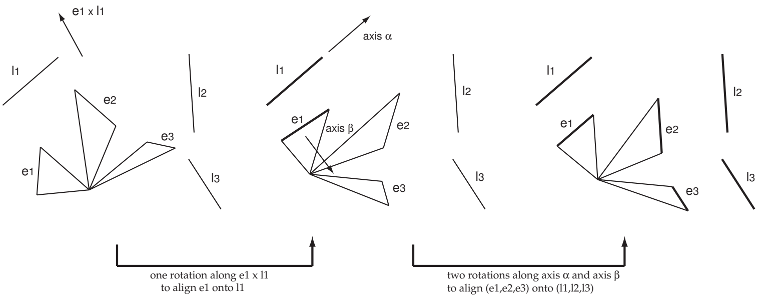


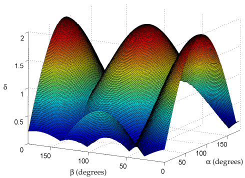
Probabilistic Edge-Line Matching

3. We draw random sets of lines.
4. For each line, we pick a random edge match.
5. For each set, we compute the camera pose and score ξ.
6. We keep the pose that minimizes ξ.
7. We initialize the correspondences using nearest-neighbor.

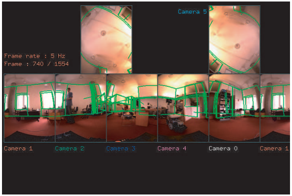


How to align three 2D edges onto three 3D lines

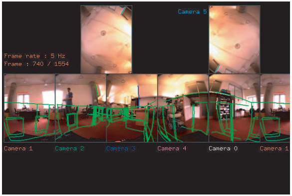




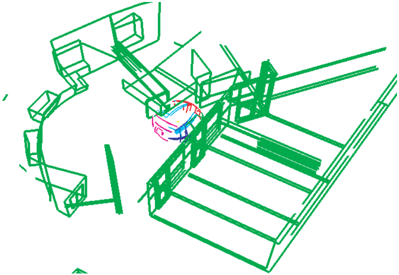
Initialization Results



Low ξ value: good candidate

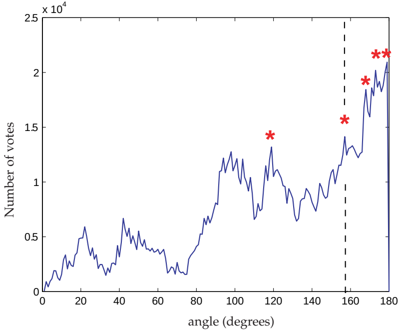


High ξ value: poor candidate



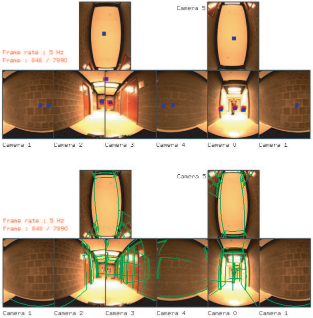
An Alternative Method: Rotation Voting

1. For each pair of image edges and each pair of model lines:
2. Compute the minimum rotation angle which brings the pair into alignment
3. Accumulate the rotation angles into a histogram.
4. Select the rotation angles with highest votes.
5. Find the most probable axis for each selected angle.



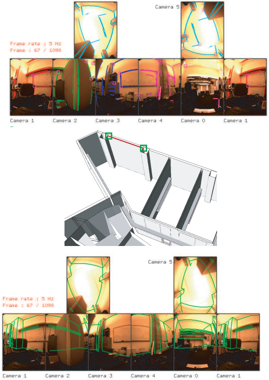
Two Alternative Methods for Initialization

- Alignment of vanishing points



Two Alternative Methods for Initialization

- Image-model corner matching



Initialization Summary

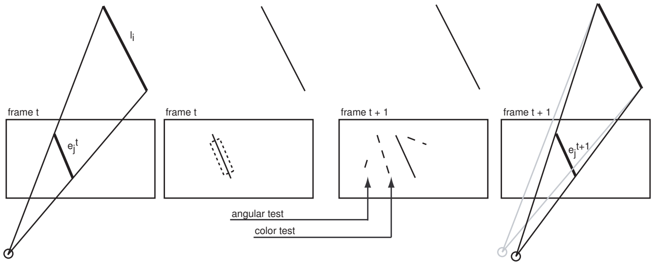
- uses a single omnidirectional image and a coarse search region
- handles strong clutter and occlusion
- based on a probabilistic line-edge matching method
- alternative methods: rotation voting, vanishing points, corner matching
- initialization in a few minutes (optimization: 10 seconds for a vertical pose)

Maintenance

We use a **multi-hypothesis** approach combined with a **hue-based filter** to maintain correspondences over successive frames.
A basic **state machine** is implemented for each correspondence in order to provide long-run consistency.

Maintenance Algorithm

1. Each correspondence is updated from frame t to frame $t+1$ using an **angular constraint** and a **hue-based** constraint.



- At the end of this process, each model line has zero, one or several candidates on the image at frame $t+1$ (**multi-hypothesis** method).

Maintenance Algorithm

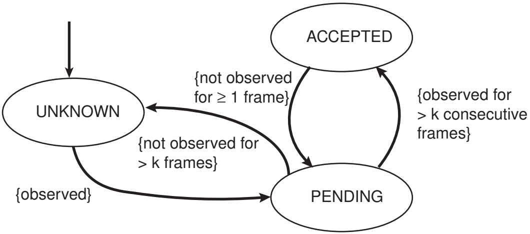
2. We draw random sets of model lines.
3. For each model line, we pick a random edge match.
4. For each set, we compute the camera pose and score ξ.
5. We keep the best camera pose and update the correspondences.

	$\lvert\bar{\mathcal{S}}_t\rvert = 30$	40	50	60
$b = 10\%$	10	9	9	8
$b = 30\%$	254	193	167	152
$b = 50\%$	29971	13743	9413	7516

Figure: Number of sample rounds needed vs. clutter percentage (b) and number of correspondences ($\lvert\bar{\mathcal{S}}_t\rvert$)

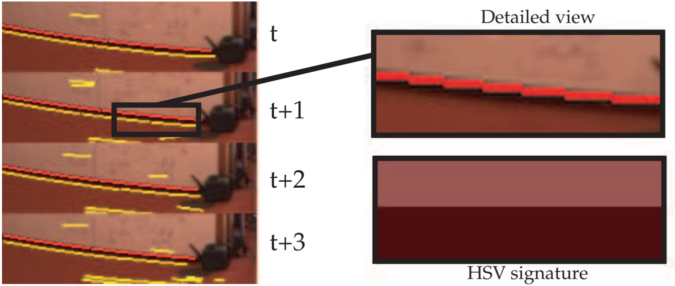
Maintenance State Machine

A state machine is implemented for each correspondence.
A correspondence starts as Unknown, upgrades to Pending when it is observed and to Accepted if it is consistently observed for several consecutive frames.



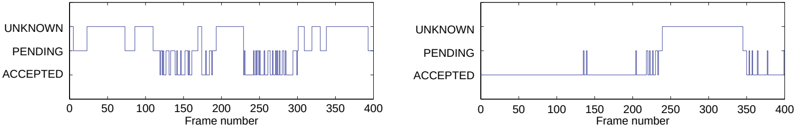
Maintenance Results

- Hue-based edge filtering



Maintenance Results

- Correspondence state machine



Maintenance Results

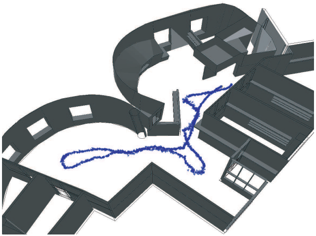
	SYNTH	LAB	CORRIDOR	HAND-HELD
MIT building number	36, 26	32-33x	36, 26	32-33x
Number of frames	100	1,500	7,800	1,900
Motion type	4-DOF	4-DOF	4-DOF	6-DOF
Excursion duration (*min*)	.33	5	26	2
Excursion length (*m*)	10	120	300	5
Walking speed (m/s)	0.5	0.4	0.20	0.04
# 3D faces in model	1900	675	1900	675
# 3D edges in model	7,400	3,000	7,400	3,000
Model surface area (m^2)	7,000	450	7,000	450
LUT size (average # faces/per node)	30	120	30	120
LUT size (average # edges/per node)	80	140	80	140

Table: Test datasets

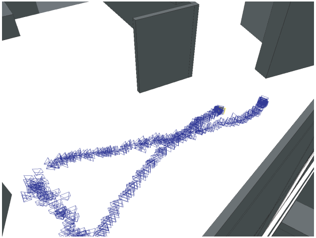
Maintenance Results

- SYNTHETIC: 6-DOF motion within a simulated lab space;
- LAB: rolling 3-DOF (x, y, θ) motion within a real lab space;
- CORRIDOR: rolling 3-DOF exploration through adjoining buildings; and
- HAND-HELD: hand-held 6-DOF motion within a real lab space.

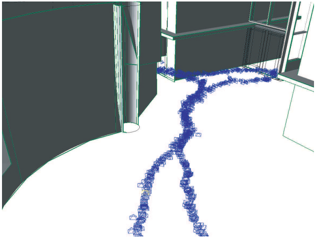
LAB dataset



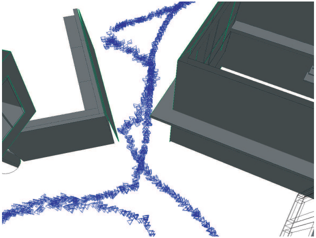
(a)



(b)

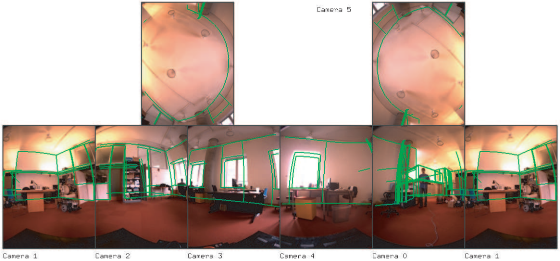


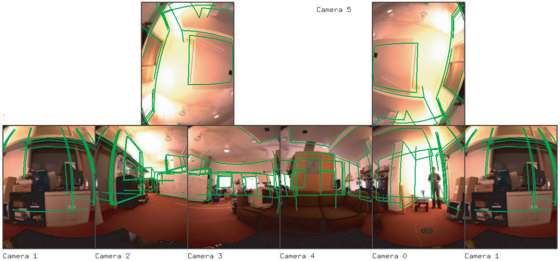
(d)



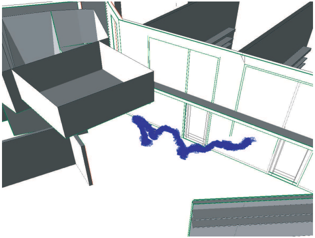
(c)

LAB dataset

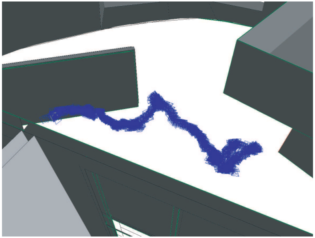

Camera 5
Camera 1
Camera 2
Camera 3
Camera 4
Camera 0
Camera 1


Camera 5
Camera 1
Camera 2
Camera 3
Camera 4
Camera 0
Camera 1

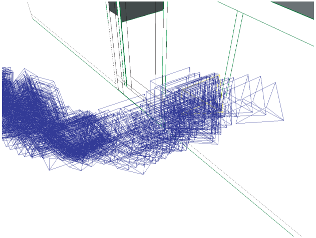
HAND-HELD dataset



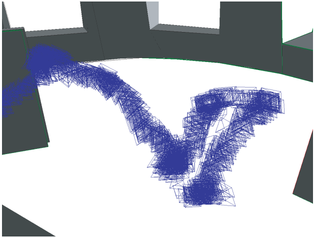
(a)



(b)

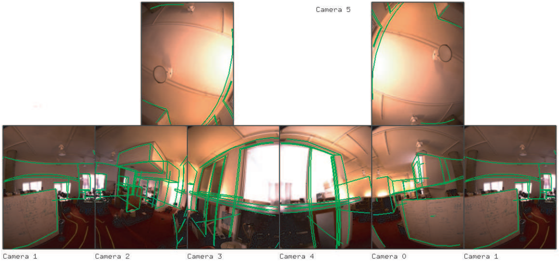


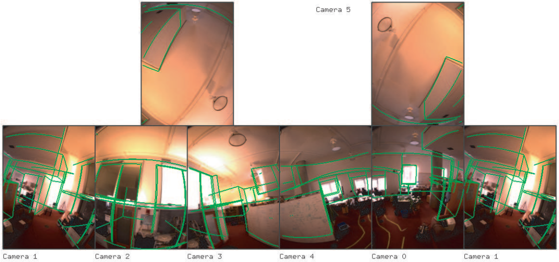
(d)



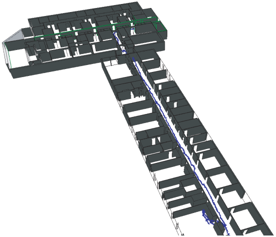
(c)

HAND-HELD dataset

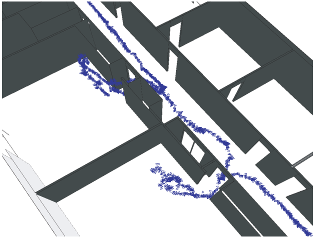




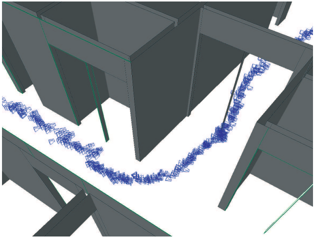
CORRIDOR dataset



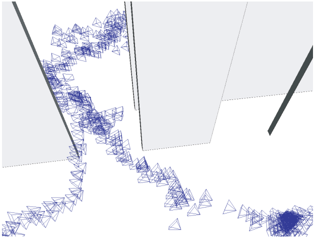
(a)



(b)

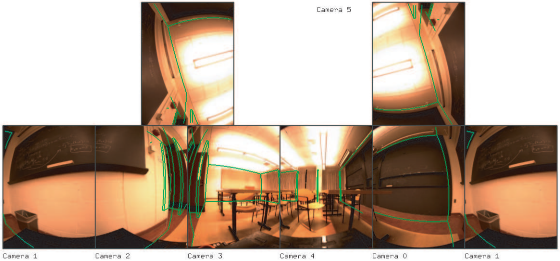


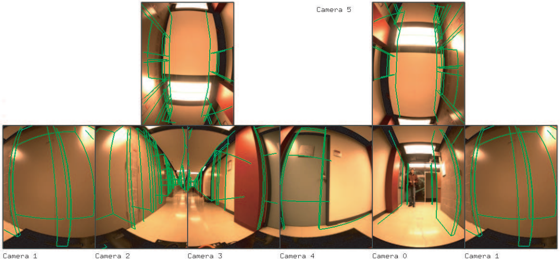
(d)



(c)

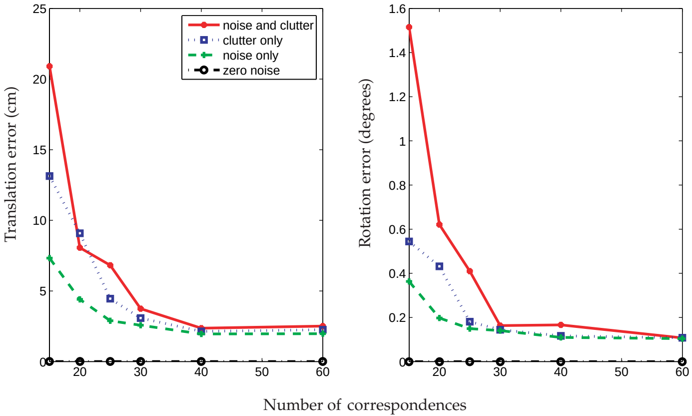
CORRIDOR dataset





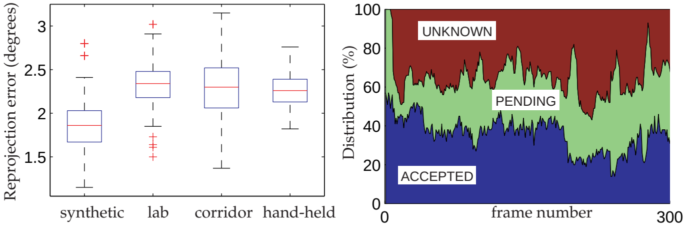
Maintenance Results

- Localization accuracy with respect to the number of correspondences.



Maintenance Results

- Re-projection error distribution for each dataset.



Conclusion

- Provides accurate and robust 6-DOF localization.
- Solves both Initialization and Maintenance.
- Scales well to large environments.
- Handle clutter and occlusion robustly.
- Makes very little assumption about the environment.
- Does not rely on a preliminary "visiting" phase.

System Limitations

- Computation requirements (1Hz).
- Requires a 3D map.
- Inaccuracies in localization : model noise, image noise, feature matching errors?
- Light-sensitivity of sensor too low.
- Initialization requires a search estimate and is sensitive to repeated environments structures (e.g. corridors).