A Self-Calibrating, Vision-based Navigation Assistant

Olivier Koch
koch@csail.mit.edu

Seth Teller
teller@csail.mit.edu

Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory (CSAIL)
Motivation

- Navigation in GPS-denied environments
  - Indoors / underground / urban areas with limited sky visibility

- Exploration and path retracing for human users
  - Soldiers in the field / Visually impaired / Disabled people
Related work

Vision-based Simultaneous Localization and Mapping

- Wolf et al., Robust Vision-Based Localization by Combining an Image Retrieval System with Monte Carlo Localization, IEEE Transactions Robotics 2005
- Davison et al., MonoSLAM: Real-Time Single Camera SLAM, PAMI 2007
- J. Neira et al., Data association in $O(n)$ for Divide and Conquer SLAM, RSS 2007
Related work

Hybrid metrical-topological localization
- Bosse et al., SLAM in Large-Scale Cyclic Environments Using the Atlas Framework, IJRR 2004
- B. Kuipers, Using the topological skeleton for scalable global metrical map-building, IROS 2004
- Zhang & Kosecka, Hierarchical Building Recognition, Image and Vision Computing 2007

Appearance-based navigation
Problem statement

Inputs:
• Training sequence
• Live video sequence

Outputs:
• Live walking guidance
• Helps user retrace path
Capture Rig & User Interface

Four IEEE1394 PointGrey Firefly Cameras
4 x 360 x 240 SIFT detection, tracking @ 4Hz
FOV: 360° (horiz.) x 90° (vert.)

Tablet PC Interface
With earphone
Novelty

- Interface:
  - Provides non-metrical walking guidance to humans
  - Guidance is “user-centered,” i.e., body-relative

- Purely vision-based:
  - Requires no camera calibration
  - Does not constrain the number of cameras or their relative positions on the capture rig
  - Novel method for correlating user motion with image feature motion
Assumptions

- User motion is smooth
- Rigid-body transformation between cameras is fixed but can change slightly over time
- System requires a brief sequence with known user motion
  - Turning slowly in place for two revolutions
  - Required only once for any given camera configuration
- Environment is 2D, mostly static and contains distinctive visual features
System Overview

- Feature matching
- SIFT detector
- Video capture rig

User interface (Visible, audible guidance)

Place graph

Rotation classifier
Capture Rig & User Interface

Four IEEE1394 PointGrey Firefly Cameras
4 x 360 x 240 SIFT detection, tracking @ 4Hz
FOV: 360° (horiz.) x 90° (vert.)

Wearable embedded PC cluster
3 x 1.8Ghz Intel Core 2 Duo CPUs
3 hours untethered operation

Tablet PC Interface
With earphone
Capture Rig & User Interface
Qualitative guidance

- Our system provides human-understandable guidance
Place Graph

- Node: place of strategic interest for navigation
- Edge: a direct physical path between nodes
- Graph built online and automatically during exploration
Navigation Guidance

- Provide rotation guidance at nodes
- Provide relative progress along edges
- In a human-understandable fashion
Place graph data structure

Node
• One image, feature set

Edge
• A series of image feature sets collected at regular time intervals
• Stored in an incremental visual dictionary [1]

Place Graph

- Graph nodes are created online and automatically:
  - Where rotation rate is high (i.e., user is turning)
  - Where scene appearance changes drastically (e.g., user exits room)

Subset of Place Graph (INDOOR dataset), with nodes overlaid manually for visualization

Example node (INDOOR dataset)
System Overview

- Feature matching
- SIFT detector
- Video capture rig
- User interface (Visible, audible guidance)

- Place graph
- Rotation classifier
Rotation classifier

field of view

Observation at time $t_1$

Observation at time $t_2$
Rotation classifier

global coordinate frame

local coordinate frame
Can we learn which rotation of the user would bring these two features in alignment with no camera calibration?
## Rotation classifier

### TRAINING

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known Motion Sequence</td>
<td>Classifier table relating user motion, feature</td>
</tr>
<tr>
<td></td>
<td>evolution in image</td>
</tr>
</tbody>
</table>

### QUERY

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two feature sets Classifier table</td>
<td>User rotation that brings the two feature sets</td>
</tr>
<tr>
<td></td>
<td>into maximal alignment</td>
</tr>
</tbody>
</table>
Rotation classifier

TRAINING

World features

User

Camera Image

Time $t_1$ Time $t_2$

Feature matching

Constant speed assumption

User at $t_1$ User at $t_2$

Coarse user rotation angle $\alpha$
Rotation classifier

TRAINING

Feature matches $t_1 - t_2$

Camera image
- Match source bin
- Match destination bin

User rotation angle $\alpha$

Match source bin
Match destination bin

Destination image

Source image
Rotation classifier

**TRAINING**

**Known motion** - user rotates in place 1 minute @ 4 Hz = 240 frames

Needs $O(n^2)$ storage:

$$n_{\text{tables}} = \binom{n}{2} + n$$

Classifier tables
Left: camera 0 – 0
Right: camera 0 – 2

Red : angle > 0
Blue : angle < 0
Rotation classifier

TRAINING

Input

Known Video Sequence
Coarse user motion

Output

Classifier table

QUERY

Input

Two feature sets
Classifier table

Output

User rotation that brings the two feature sets into maximal alignment
**Navigation at node**

First visit \((t = t_1)\)

Revisit \((t = t_2)\)

**Method**

*Input*: Observation at time \(t_1\) (visit) and time \(t_2\) (revisit)

*Method*: Match features between \(t_1\) and \(t_2\) observation

*Output*: For each match, query the classifier and return a rotation angle \(\alpha\)

Run RANSAC voting to determine optimal rotation angle \(\alpha\)

Rotation guidance to align user with appropriate outgoing edge
Navigation along edges

Input

A series of observations $S_0 = \{o^1, \ldots, o^n\}$ along edge (first visit)

Current observation $o^t$

Output

Relative progress along the edge (normalized from 0 to 100%)
Navigation along edges

Method: recursive state estimator

State vector $\mathbf{v}$.

$\mathbf{v}_i$ represents the probability of the user standing at location of observation $o^i$.

Initialization (user leaving node)

$\mathbf{v}_i = 1$ if $i=0$ (i.e., user is at start of edge), 0 otherwise.
Navigation along edges

At each time step, given a new observation $o^t$:

\begin{align*}
\nu^t & \xrightarrow{\text{transition}} \tilde{\nu}^{t+1} & \tilde{\nu}^{t+1} & \xrightarrow{\text{observation}} \nu^{t+1}
\end{align*}

- **Transition update** (motion continuity assumption)
  
  \[ \tilde{\nu}^{t+1} = \nu^t \otimes \text{Gaussian} \ (0, \sigma) \]

  where $\sigma$ is a function of frame rate and typical user motion speed

- **Observation update**
  
  \[ \nu^{t+1}_i = \tilde{\nu}^{t+1}_i \times P(o^i, o^t) \]

  where $P(a, b)$ is the probability that $a$ and $b$ are observed from the same location
## Datasets

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration</th>
<th>Path length</th>
<th>Frame rate</th>
<th># frames</th>
<th># nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDOOR</td>
<td>45 min</td>
<td>~2.5 km</td>
<td>4 Hz</td>
<td>11,000</td>
<td>280</td>
</tr>
<tr>
<td>OUTDOOR</td>
<td>12 min</td>
<td>~1 km</td>
<td>4 Hz</td>
<td>2,900</td>
<td>43</td>
</tr>
</tbody>
</table>

**INDOOR Dataset**
MIT Tunnel network

**OUTDOOR Dataset**
Kendall Square, Cambridge MA
Rotation guidance at nodes

Fig. 1 - Rotation guidance output while user rotates in place in a new environment

Fig. 2 – Error distribution against IMU-ground truth. **Standard deviation = 12 deg.**
Progress guidance along edges

Belief state propagation while user walks along an edge (INDOOR dataset)

Relative progress along several consecutive edges. Ground truth estimated using constant speed assumption. Std. dev. is 3.3 frames (1 second, ~1.5m)
Topological map automatically generated by the system (INDOOR dataset). Nodes manually overlaid on map for visualization.
Failure modes

- Ambiguous configurations
- User leaves the exploration path
- Highly repetitive environments (featureless corridors)
- Significant change in lighting
- Dynamic scenes (e.g., crowds)
- Fast user motion (motion blur) or low lighting
Future work

- Global localization
- Extend to 3D motion
- Path self-intersection (non-linear graphs / loop closure)
- Augmented reality applications (e.g., in situ virtual tagging)
Summary

Inputs:
- Training sequence
- Video sequence

Outputs:
- Loose guidance in 2D
- Supports user retrace

- Requires no intrinsic or extrinsic camera calibration
- Generalizes to any number / configuration of cameras
  - Requires roughly fixed rigid-body transform between cameras
- New way of correlating user motion and image motion
- Provides loose guidance / directions to user
Discussion