# Vision-based Mobile Robot Localization and Mapping using Scale-Invariant Features

Stephen Se, David Lowe, Jim Little
Department of Computer Science
University of British Columbia

Presented by Adam Bickett

## Objective: SLAMB

- **S**imultaneous **L**ocalization **a**nd **M**ap **B**uilding
  - Need localization to build a map
  - Need a map to accurately localize
  - Critical for successful robot navigation in a large environment

## Previous Approaches

- Different sensor modalities:
  - Sonar
  - Lasers
  - Vision
- Many early approaches require beacons or visual patterns present in the environment
- How do we use vision techniques in an unmarked environment?

## Previous Approaches

- Most previous approaches allow significant drift over time
  - DROID system uses image corner features, tracked with Kalman filters. [Harris]
  - [Murray et al] track features with a 2D occupancy map, but localization is not updated, so odometry errors accumulate.

## Previous Approaches

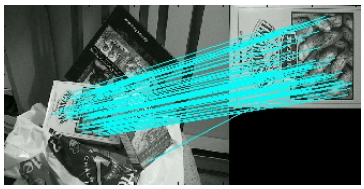
- Other approaches use probabilistic modeling
  - Batch EM approach which estimates the most likely map by considering all previous locations of sonar scans [Thrun et al]
  - Monte Carlo localization [Dellaert et al] updates probability distribution of current pose given new sensor input
  - Most of these use laser or sonar.

## Using SIFT for SLAMB

- Much of the previous work uses laser or sonar based sensors, why not use vision?
  - More processor intensive
  - Stable vision features are hard to extract
- Approach: Build a 3D map of SIFT features, constructed using trinocular stereo rig mounted on the robot

## SIFT features

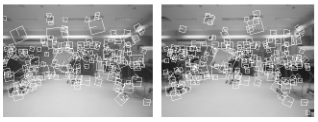
- SIFT feature attributes make them well-suited for the purpose
  - Invariant to image translation, scaling, rotation
  - Partially invariant to illumination and affine/3D projections
  - Keypoints are at maxima/minima of DoG applied at different scales.



## Using SIFT features

- Matching between SIFT features in the three views provides stable landmarks for the environment

(a)



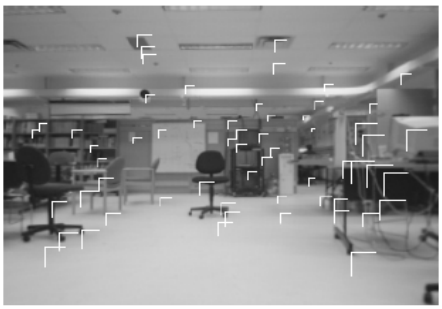
## Stereo Matching with SIFT

- The right camera is used as a reference camera
  - Left camera is 10 cm left, top camera 10 cm up
- Constraints:
  - Epipolar constraint
  - Disparity constraint
  - SIFT scale/orientation constraint
- If a feature has more than one match, it is discarded as ambiguous

## SIFT Stereo Matching

- From the final matching set, the 3D world coordinates (X,Y,Z) can be computed
- Each match can be stored with coordinates and averaged SIFT orientation/scale.
- Relaxing some constraints doesn't necessarily increase matches because ambiguous points are discarded

## Example Matches



(a)

## Ego-motion Estimation

- For each matched feature, calculate:

$$[r, c, s, o, d, X, Y, Z]$$

  - (r,c): measured image coordinates in reference camera
  - (s,o,d) : scale, orientation, disparity for each feature
  - (X,Y,Z) : estimated real-world coordinates
- To build a map, movement measurements between scenes is needed
- Odometry data is only a rough estimate
- Use point correspondences in least-squares estimation to calculate a more accurate localization

## Limiting the feature-matching search

- Searching the whole database for SIFT feature matches is too expensive
  - Use the odometry data to compute the expected location and scale of features in the new scene
  - The search for SIFT feature matches is constrained to a 5x5 pixel window around this estimate.

## Results – Matching across movement

a) 10 cm forward movement

b) 5 degree rotation

| | No. of matches | % of matches |
|---|---|---|
| a) | 43 | 73% |
| b) | 41 | 68% |

## Least-Squares Error Minimization

- To refine the initial pose (R,T) estimates between the matched points , use Newton's method:

$$\boldsymbol{p}^{(i+1)} = \boldsymbol{p}^{(i)} - \boldsymbol{x}.$$

$$\boldsymbol{J}\boldsymbol{x} = \boldsymbol{e}$$

$$J_{ij} = \frac{\partial e_i}{\partial x_j}.$$

$$\min ||\boldsymbol{J}\boldsymbol{x} - \boldsymbol{e}||^2.$$

$$\boldsymbol{J}^T\boldsymbol{J}\boldsymbol{x} = \boldsymbol{J}^T\boldsymbol{e}.$$

$p^i$ :parameters at step I

*x :* terms to subtract to minimize error

*J*: Jacobian matrix encoding derivative of error wrt change in parameter

One observed error $e_i$ should be equal to the sum of changes of that error w.r.t changes in the parameters *x*

Solve the resulting system of linear equations

## Pose Estimation Results

- Odometry measurements are used to initialize the minimization
- SIFT features give very high percentage of inliers; so outliers are discarded (residual error >3 pixels)
- We're estimating the 6 d.o.f. in (T,R)

| Fig | Odometry | Mean $E$ | Least-Squares Estimate |
|---|---|---|---|
| 3(a) | Z=10cm | 1.328 (pixels) | [1.353cm,-0.534cm,11.136cm, $0.059^\circ, -0.055^\circ, -0.029^\circ$] |
| 3(b) | $\theta$=5° | 1.693 (pixels) | [0.711cm,0.008cm,-0.9890cm, $4.706^\circ, 0.059^\circ, -0.132^\circ$] |

## Landmark Tracking

- Need a database to store observed SIFT landmarks for subsequent matching

$$[X, Y, Z, s, o, l]$$

  - Entry includes position w.r.t camera, orientation, scale of SIFT feature, and *l*, consecutive frames feature has been missed
  - Features not seen for N (20) frames, and expected to be in view, are pruned.
  - New features are added after 3 frames
  - Matched features values are averaged, and these features are used to update pose estimation.

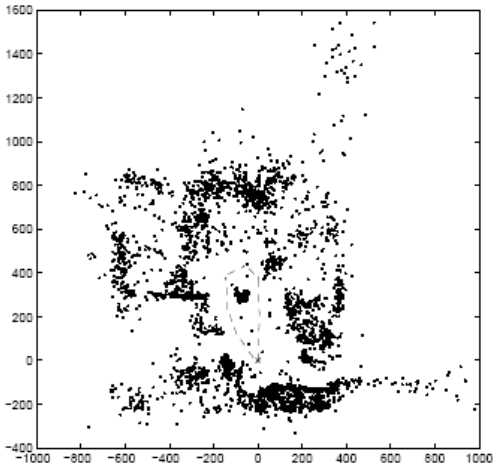
## Experimental Results

$$\text{X:-2.09cm Y:0cm Z:-3.91cm}$$
$$\theta{:}0.30^\circ\ \alpha{:}2.10^\circ\ \beta{:}-2.02^\circ$$

Estimated position after a (~8m) tour of the lab

- Change in height is forced to 0
- Error minimization terms are limited; odometry between frames is fairly accurate
- 40-60 matches per frame
- 3590 SIFT features in database at the end
- Runs at 2Hz on a PIII

## SIFT feature map




## Enhancements – Permanent Landmarks

- In a volatile environment, we need a way to know which features to keep
  - Keep features which meet a certain reliability: percentage of frames they are matched out of the total frames they are expected
  - Create a static database of reliable features while environment is clear.
  - These features won't be pruned when occluded

## Enhancements – Viewpoint Variation

- Even though SIFT features are reasonably invariant, they can't accommodate large changes in viewpoint
  - Store view angle with SIFT feature
  - If current view angle of a matched feature is larger than a threshold (20°), add the SIFT feature from new viewpoint to the same landmark entry

## Questions?