Development of a Robust Indoor 3D SLAM Algorithm

Timothy Murphy
Honors Tutorial College
Dr. David Chelberg

Ohio University
School of Electrical Engineering and Computer Science
Russ College of Engineering and Technology

Introduction

- Problem:
  - Create Map during Exploration of Environment
  - Must Know Location to Accurately Create Map
  - Must have a map to accurately find location.
- Applications
  - Search and Rescue
  - Home Health
- Simultaneous Localization and Mapping (SLAM)
  - Appearance Based SLAM
  - Depth Based SLAM
- Turtlebot 2.0
  - Sensors:
    - Microsoft Kinect
    - Inertial Unit
    - Bump and Cliff Sensors
  - Laptop: Asus X200-CA
    - Celeron 100070U @ 1.3 GHz
    - 4 GB RAM
    - 320 GB Hard Drive


from www.turtlebot.com

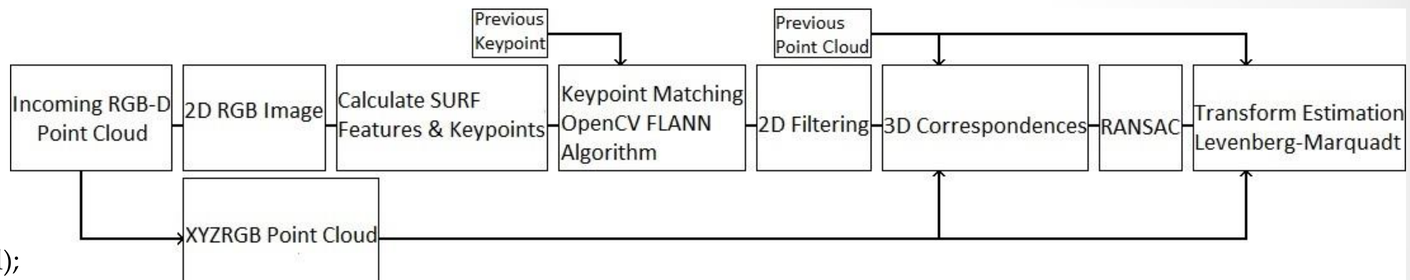
3D SLAM Algorithms

- 3D SLAM
  - Appearance Based SLAM
    - Use Color Features
  - Depth Based SLAM
    - Uses Depth Features

- Our Algorithm
  - Calculates Color Features
  - Combine Color Features with Depth Information
  - Find Matching Keypoints of Color and Depth Features
  - Calculate Transformation based on Keypoints.

Outline of Algorithm

- Input: Sequential RGB-D Point Clouds
- Output: 3D Point Cloud Aligned into single frame




```
3D_SLAM{
  Clouds = {};
  Keypoints = {};
  Descriptors = {};
  Transform = I
  while(!user.quit()){
    cloud = get_next_cloud();
    Clouds.append(cloud);
    img = get_2d_image(cloud);
    feature, keypoint = get_2D_keypoints(img);
    Keypoints.append(keypoint);
    Descriptors.append(feature);
    if (length(Clouds) > 1){
      2D_corr = get_2D_correspondences(Keypoints[-1], Descriptors[-1],Keypoints[-2], Descriptors[2]);
      3D_corr = get_3D_correspondences(2D_corr, Clouds[-1], Clouds[-2]);
      RANSAC(3D_corr);
      Transform = get_transform(3D_corr);
      Clouds[-1] *= Transform;
    }
  }
}
```

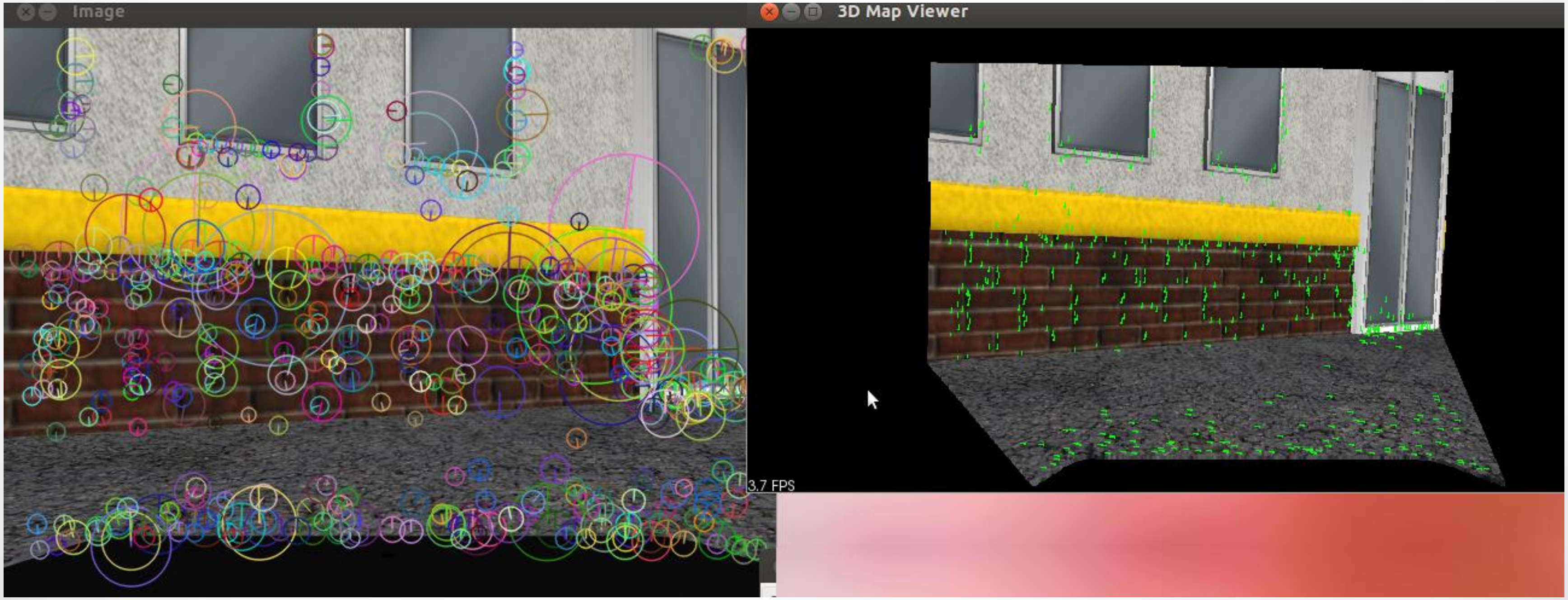
Feature Extraction

- Features
  - Image Structures: Points, Edges, Corners, Objects
  - Local properties of an image
  - Invariant to Translation, Scale, Rotation
- Importance
  - Improve Performance
  - Allows comparison of Images
  - Should be invariant to Transforms

Surf Features (from www.opencv.com)

SURF Features

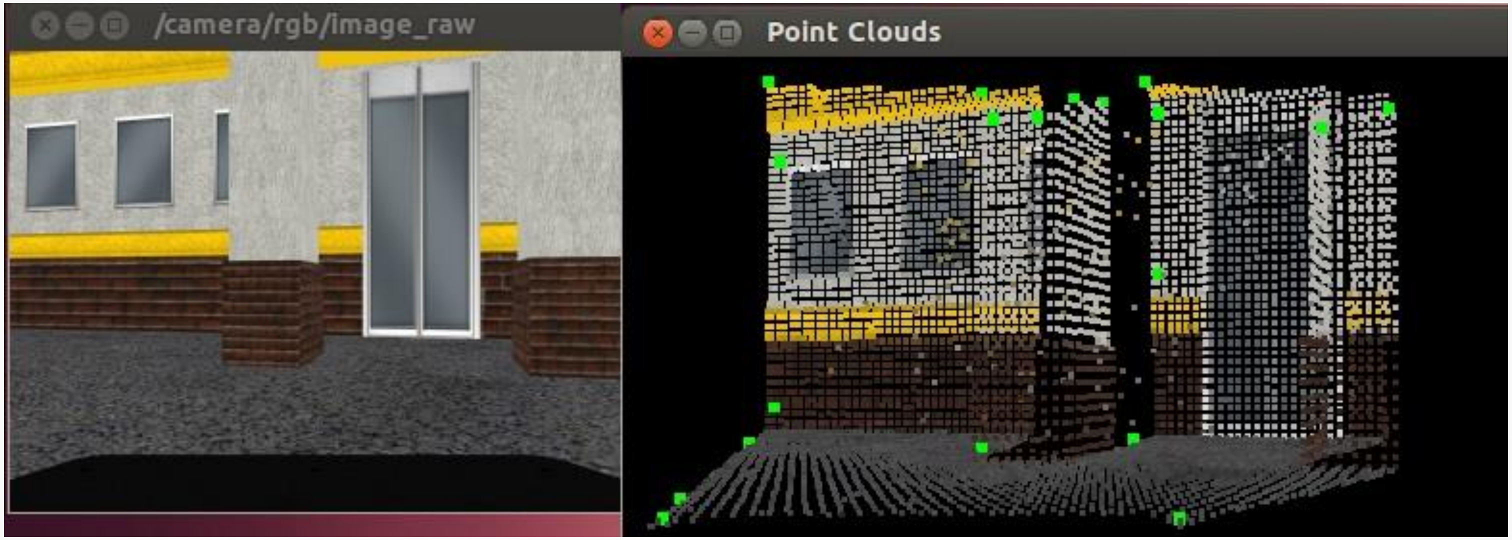
- Speeded Up Robust Features (SURF) – OpenCV
  - Local Feature Detector and Keypoint Finder
  - Uses Hessian Blob Detector
    - Affine-invariant feature detector based on second partial derivatives of image smoothed using a Gaussian Kernel



Surf Keypoints 2D image (left) SURF Keypoints shown in 3D Coordinate Space (right).

NARF Features

- Normal Aligned Radial Features (NARF) – PCL
  - Local Feature Detector and Keypoint Finder
  - Calculates change of normal around points of interest using depth image.



2D Image (left) Down Sampled Point Cloud & extracted NARF Features (right)

- SURF vs NARF
  - SURF Feature Extraction takes 0.133 Seconds
  - NARF Feature Extraction takes 14.405 Seconds
  - SURF Features produces more Keypoints that were more stable.

Keypoint Correspondences

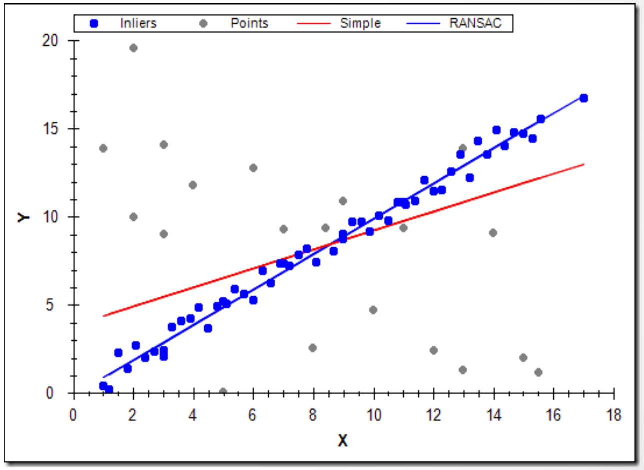
- Calculate Corresponding Keypoints
  - OpenCV's Fast Library for Approximating Nearest Neighbors (FLANN)
- Filter Results using Threshold
  - All transforms between images are Affine (Scale, Translation, Rotation)
  - Distance between corresponding keypoints should be similar
  - Reject any Correspondences that are too far apart
- Remaining Keypoints are Associated into 3D Space
  - Reverse Process of calculating 2D Image from 3D point Cloud

SURF Keypoint Correspondences

SURF Keypoint Matches in 2D Image. Incoming Image (left) Previous Image(right)

RANSAC

- Random Sample Consensus (RANSAC)
  - Iterative Method for estimating parameters of a Mathematical Model
    - Select random sample of data points to fit model
    - Test remaining points if they fit the model.
  - Produces a set of Inliers and Outliers
    - Inliers are the data points that fit the model
    - Outliers are the points that do not
  - Try to maximize the set of Inliers



from http://crsouza.blogspot.com/

- Use RANSAC on 3D Correspondences
  - Reject the outlier set
  - Remove poorly matched Correspondences

Transformation Estimation

- Levenberg Marquardt Transformation Estimation
  - Non-linear minimization of least-square cost function
    - Distance between corresponding keypoints
  - Iterative method for finding Rigid Transform Estimation

- Singular Value Decomposition
  - Used to minimize least squares of cost function
  - Can have closed form solution to find Rigid Transformation Matrix

Further Improvements

- Additional Keypoint types
  - BRISK, SIFT, FAST keypoints
  - Use Multiple Keypoint types
  - Determine which keypoint type works best in current environment
- Improve Levenberg Marquardt method
  - Alter implementation to allow to test for convergence of point clouds
  - Calculate how well aligned the resulting transformation is
- Loop Closure
  - Allows the robot to know it has already seen the area before
  - Adds additional constraints to the map to improve errors in map
- Increase robustness, reliability and the quality of the generated Map for future navigation purposes.

Conclusion

- Current results produce a 3D Map that combines all input Point Clouds.

- These Results can be improved for better performance and accuracy of the Map.

Questions

References

- [1] Bay, H., Ess, A., Gool, L., Tuytelaars, T. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU).* (2008). Vol 110, No. 3, pgs 346-359.
- [2] Cheeseman, P., Self, M., Smith, R. Estimating Uncertain Spatial Relationships in Robotics. *Proceedings of the Second Annual Conference of Uncertainty in Artificial Intelligence*. (1986). University of Pennsylvania, Philadelphia, PA, USA: Elsevier. pgs 435-461.
- [3] Fox, D., Henry, P., Herbst, E. Krainin, M., Ren, X. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. Web. Viewed 21 April 2014. http://www.cs.washington.edu/robotics/postscripts/3d-mapping-iser-10-final.pdf. 2013.
- [4] Flannery, B., Press, W., Teukolsky, S., Vetterling, W. Numerical Recipes 3rd Edition: The Art of Scientific Computing. *Columbia Press*. (September 2007).
- [5] Hung, D., Sun, C., Wang, Y. Improving Data Association in Robot SLAM with Monocular Vision. *Journal of Information Science and Engineering 27*. (March 2011). pgs 1823-1837.
- [6] Labbe, M., Michaud, F. Appearance-Based Loop Closure Detection for Online Large-Scale and Long-Term Operation. *IEEE Transactions on Robotics*. (June 2013). pgs 734-745
- [7] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. (2004). Vol 60, No. 2, pgs 91-110.
- [8] Mikolajczyk, k., Schmid, C. An affine invariant interest point detector. *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*. 2002.
- [9] Mirowski, P. Depth Camera SLAM on a Low-cost WiFi Mapping Robot. Web. Viewed 21 September 2014. http://www.academia.edu/2527447/Depth_camera_SLAM_on_a_low-cost_WiFi_mapping_robot
- [10] Description of Turtlebot 2.0 can be found at http://www.turtlebot.com/
- [11] Description of Microsoft Kinect and how it can be used can be found at http://www.microsoft.com/en-us/kinectforwindows/meetkinect/features.aspx
- [12] For specifics on OpenCV, PCL, and ROS visit http://www.opencv.org, http://pointclouds.org, and http://www.ros.org respectively.
- [13] http://crsouza.blogspot.com/2010/06/random-sample-consensus-ransac-in-c.html