Indoor Space Mapping using Kinect XBOX360 and RTAB-Map

Final project in course “Introduction to robotics”

Nora Nikoloska
Faculty of Computer Science and Engineering
Skopje, North Macedonia
nora.nikoloska@hotmail.com

Monika Mateska
Faculty of Computer Science and Engineering
Skopje, North Macedonia
monikamateska@hotmail.com

***Abstract*—This paper presents a 3D map creation of our dean office of the Faculty using Kinect XBOX 360 and Real Time Appearance Based Mapping software (RTAB-Map) for processing the scan. The software mapping is based on the simultaneous localization and mapping (SLAM) technique.**

Keywords—mapping, SLAM, RTAB-Map

Introduction

In this project the following tools were used:

Hardware:

- Kinect XBOX 360 (provided by the faculty)

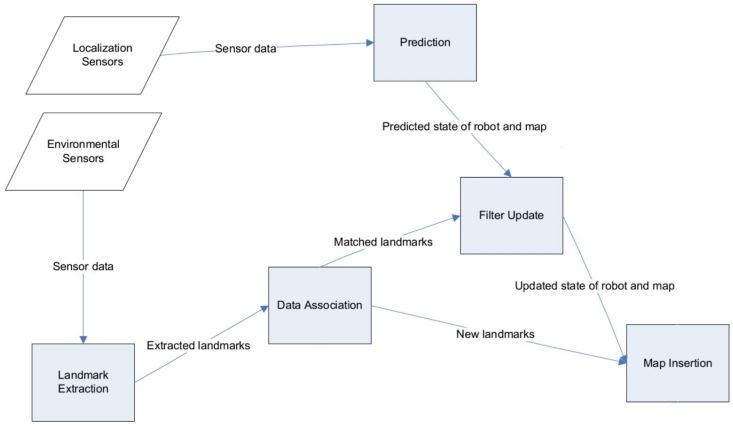
Software:

- 64-Bit Windows 10 Desktop
- Kinect for Windows Software Development Kit (SDK) 2.0
- ROS (Robot Operating System)
- RTAB-Map
- MeshLab

RTAB-Map is based on a loop closure approach to determine how likely a new image is already seen or if it is a new location. When the same hypothesis is accepted, the errors of the constructed map are minimized while creating a bigger picture of the environment. We used a Kinect XBOX 360 camera in order to scan the location with the help of its sensors.

I. Description of SLAM algorithm

The SLAM technique enables us to compute the robot’s pose and the map of the environment, both at the same time, i.e. do localization and mapping simultaneously.
Because both, the map and the pose need correlation, it is hard to define the one term value without previously obtaining the other.



Extended Kalman Filter is one of the algorithms used in this project. It estimates the state of a dynamic system, given the model of the system, model of the sensors, measurements value with noise from the sensor as well as the control inputs.

It is divided into two steps, prediction and correction.

$$\text{System state: } \bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathcal{R}} \\ \bar{\mathcal{M}} \end{bmatrix} = \begin{bmatrix} \bar{\mathcal{R}} \\ \bar{\mathcal{L}}_1 \\ \vdots \\ \bar{\mathcal{L}}_n \end{bmatrix}$$

Covariances matrix:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \\ \mathbf{P}_{\mathcal{M}\mathcal{R}} & \mathbf{P}_{\mathcal{M}\mathcal{M}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{R}\mathcal{L}_n} \\ \mathbf{P}_{\mathcal{L}_1\mathcal{R}} & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_1\mathcal{L}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{\mathcal{L}_n\mathcal{R}} & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_1} & \cdots & \mathbf{P}_{\mathcal{L}_n\mathcal{L}_n} \end{bmatrix}$$

Goal: keep the map $\bar{\mathbf{x}}$, $\mathbf{P}$ up to date at all times

1. The prediction is based on:

   Movement function (part of the system model):
   $\mathbf{x} \leftarrow f(\mathbf{x}, \mathbf{u}, \mathbf{n})$
   - $\mathbf{u}$ ... control vector
   - $\mathbf{n}$ ... pertubation vector
   - $p(\mathbf{n}) \sim N(0, \mathbf{Q})$

   EKF prediction step in SLAM:

$$\bar{\mathcal{R}} \leftarrow f_{\mathcal{R}}(\bar{\mathcal{R}}, \mathbf{u}, 0)$$
$$\mathbf{P}_{\mathcal{R}\mathcal{R}} \leftarrow \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{R}} \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}}^{\mathsf{T}} + \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}} \mathbf{Q} \frac{\partial f_{\mathcal{R}}}{\partial \mathbf{n}}^{\mathsf{T}}$$
$$\mathbf{P}_{\mathcal{R}\mathcal{M}} \leftarrow \frac{\partial f_{\mathcal{R}}}{\partial \mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{M}}$$
$$\mathbf{P}_{\mathcal{M}\mathcal{R}} \leftarrow \mathbf{P}_{\mathcal{R}\mathcal{M}}^{\mathsf{T}}$$

2. Correction step is covering the re-calculation:

   Observation function (part of the sensor model): $\mathbf{y} = h(\mathbf{x}) + \mathbf{v}$
   - $\mathbf{v}$ ... meassurement noise
   - $p(\mathbf{v}) \sim N(0, \mathbf{R})$

   EKF correction step in SLAM for every landmark $\mathcal{L}_i$:

$$\bar{\mathbf{z}} = \mathbf{y}_i - h_i(\bar{\mathcal{R}}, \bar{\mathcal{L}}_i) \text{ 'innovation'}$$
$$\mathbf{Z} = \begin{bmatrix} \mathbf{H}_{\mathcal{R}} & \mathbf{H}_{\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_i} \\ \mathbf{P}_{\mathcal{L}_i\mathcal{R}} & \mathbf{P}_{\mathcal{L}_i\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{\mathcal{R}}^{\mathsf{T}} \\ \mathbf{H}_{\mathcal{L}_i}^{\mathsf{T}} \end{bmatrix} + \mathbf{R}$$
$$\mathbf{K} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_i} \\ \mathbf{P}_{\mathcal{L}_i\mathcal{R}} & \mathbf{P}_{\mathcal{L}_i\mathcal{L}_i} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{\mathcal{R}}^{\mathsf{T}} \\ \mathbf{H}_{\mathcal{L}_i}^{\mathsf{T}} \end{bmatrix} \mathbf{Z}^{-1} \text{ 'Kalman gain'}$$
$$\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} + \mathbf{K}\bar{\mathbf{z}}$$
$$\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\mathbf{Z}\mathbf{K}^{\mathsf{T}}$$

   - $\mathbf{H}_{\mathcal{R}} = \frac{\partial h_i(\bar{\mathcal{R}}, \bar{\mathcal{L}}_i)}{\partial \mathcal{R}}$, $\mathbf{H}_{\mathcal{L}_i} = \frac{\partial h_i(\bar{\mathcal{R}}, \bar{\mathcal{L}}_i)}{\partial \mathcal{L}_i}$

Last, but not least is the map insertion, which represents the matrix created by calculating the of the values of the previous two steps, the prediction and the correction.

$$\mathcal{L}_{n+1} = g(\bar{\mathcal{R}}, \mathbf{y}_{n+1})$$
$$\mathbf{G}_{\mathcal{R}} = \frac{\partial g(\bar{\mathcal{R}}, \mathbf{y}_{n+1})}{\partial \mathcal{R}}$$
$$\mathbf{G}_{\mathbf{y}_{n+1}} = \frac{\partial g(\bar{\mathcal{R}}, \mathbf{y}_{n+1})}{\partial \mathbf{y}_{n+1}}$$
$$\mathbf{P}_{\mathcal{L}\mathcal{L}} = \mathbf{G}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{R}} \mathbf{G}_{\mathcal{R}}^{\mathsf{T}} + \mathbf{G}_{\mathbf{y}_{n+1}} \mathbf{R} \mathbf{G}_{\mathbf{y}_{n+1}}^{\mathsf{T}}$$
$$\mathbf{P}_{\mathcal{L}\mathbf{x}} = \mathbf{G}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathbf{x}} = \mathbf{G}_{\mathcal{R}} \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \end{bmatrix}$$
$$\bar{\mathbf{x}} \leftarrow \begin{bmatrix} \bar{\mathbf{x}} \\ \mathcal{L}_{n+1} \end{bmatrix}$$
$$\mathbf{P} \leftarrow \begin{bmatrix} \mathbf{P} & \mathbf{P}_{\mathcal{L}\mathbf{x}}^{\mathsf{T}} \\ \mathbf{P}_{\mathcal{L}\mathbf{x}} & \mathbf{P}_{\mathcal{L}\mathcal{L}} \end{bmatrix}$$

II. Kinect Scanning

To conduct a 3D scan of the offices we used the RTAB-Map software. In order to connect to our Kinect sensor using the OpenNI2 driver, we installed the Windows SDK for Kinect. In the RTAB-Map GUI we used the 3D Map view, the Loop closure detection view and the Odometry view. It is necessary to monitor the odometry because if it is lost and the screen turns red, we must reset it using the Detection -> Reset Odometry option. If the detection is lost, the local map cannot connect to the global scan of the room.

The loop closure detection checks if the current frame is already familiar to the scan. When starting a new scan, it is best to start with familiar objects in order for the detection algorithm to localize the current map.

One obstacle that we experienced is that the longer the scan is and the number of frames for loop detection is increased, the computational time for the detection is larger and may result in the program being unresponsive for some time.

Another advice for a better scan is to avoid plain surfaces without any features like white walls or windows, since the software is using most of the distinct features of the image for detection.

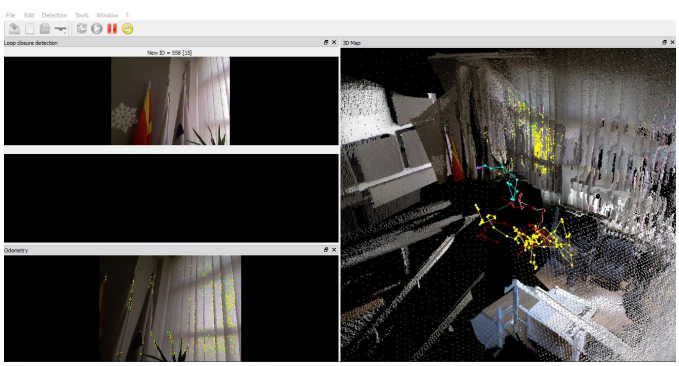


Image 1: Indoor Kinect Scanning with RTAB-Map

III. Transforming the point cloud to mesh

RTAB-Map export options include the option to export the scan in the form of a point cloud.

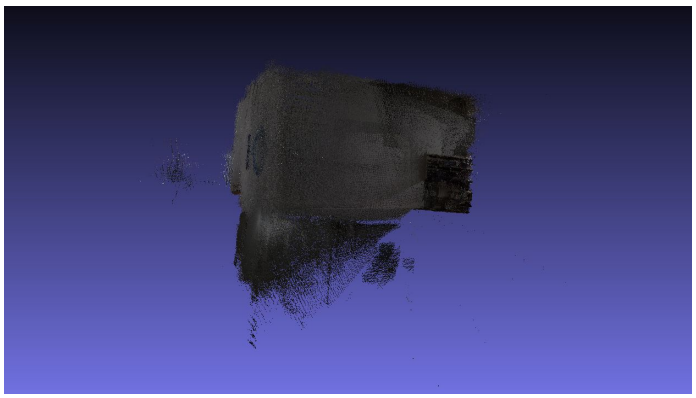


Image 2: Point cloud of the scan

However, upon obtaining the point cloud as a .ply file, we had to transform it into a solid object.

For the process of transforming the point cloud obtained with the Kinect room screening into a solid mesh we used MeshLab. The steps for this process are the following:

1. Importing the mesh - .ply file of the point cloud
2. Subsampling the points

In order to decrease computation time, subsampling is applied by using the poisson - disk subsampling option located in the Filter -> Subsampling -> Poisson Disk Subsampling menu. In this stage, the parameters used may differ according to the type of the scan. The parameters we used for indoor scan that differ from the defaults are the following:

- Number of samples: 250000
- MonteCarlo OverSampling: 200
- Base Mesh Subsampling

3. Reconstructing the normals

With this step we assign a direction to each point to be facing either 'in' or 'out' of the mesh. The option used is in the Filter -> Point Set -> Compute the normals menu. The parameters we used are the following:

- Number of neighbours: 1000
- Flip normals

4. Surface reconstruction

This is the final step in the process that constructs the solid mesh. The option used to create a surface is in the Filter -> Remeshing, Simplification and Reconstruction -> Screened Poisson Surface Reconstruction. This step takes a lot of time to compute the mesh if the point set is too large, which is why the subsampling step is necessary.

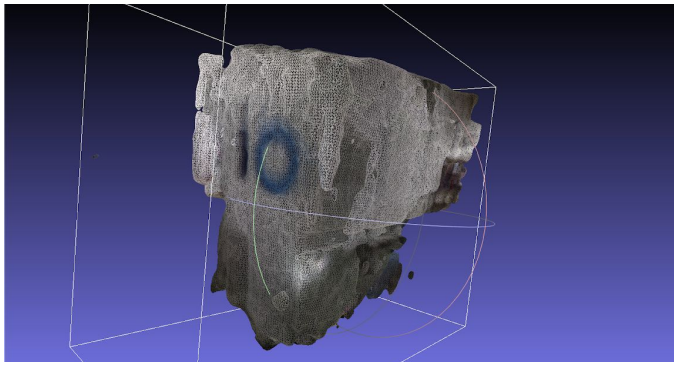


Image 3: Solid mesh of the dean offices

To this mesh we can apply different colors, textures and shadings.

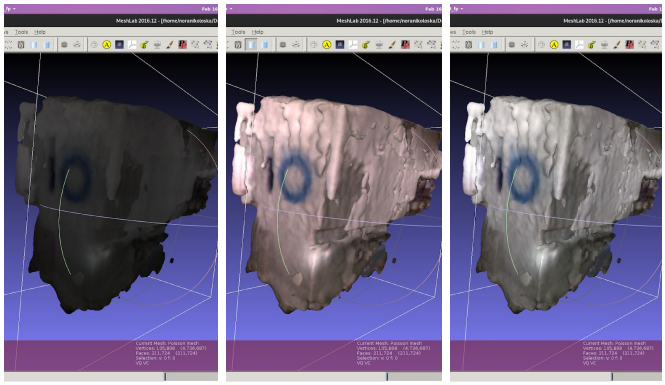


Image 4: Different values for black - face

Acknowledgment

We thank the Faculty of Computer Science and Engineering - Skopje for providing the Kinect XBOX 360 Sensor and indoor mapping space.

References

[1] W. Burgard, C. Stachniss, K. Arras, M. Bennewitz "Simultaneous Localization and Mapping" .

[2] Juan-Antonio Fernández-Madrigal, J. Luis Blanco Claraco "Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods" , September, 2012.

[3] A. Frischenschlager "SLAM Algorithm Simultaneous Localization And Mapping" , December 17 2013, pg. 8-26.