HW2 Report

Guosong Li(A53277848)

1. Introduction

SLAM(Simultaneous Localization and Mapping) refers to a set of problem of trying to simultaneously localize the position with respects to its surroundings using noisy sensors, and meanwhile mapping the structure of the environment. This is a chicken-and-egg problem because intelligent system cannot map the environment around it without knowing its location.

SLAM is one of the key driving forces to autonomous vehicle, drones and robotics. The main advantage of SLAM over many similar approaches of recovering pose and structure lies on that it can operate in real time, meaning that the process is done before sensor data comes in next time step. The reason why SLAM is so critical to these real time applications is that they can quickly response to a new environment and thus provide an opportunity of making a better decision on further movements.

More recently the use of visual sensors has become an important aspect of SLAM research, in part because an image provides a rich source of information about the structure of the environment (containing more information than a sonar or lidar for example). A large amount of research on visual SLAM used stereo cameras, or cameras alongside other sensors (such as accelerometers or GPS). But since around 2001 several works showed how SLAM could be done successfully using only a single camera (known as problem visual SLAM).

This report will implement particle filter and occupancy grid map with differential-drive motion model and map correlation observation model.

2. Problem Formulation

Solutions to the SLAM problem exploit the decomposition of the joint pdf due to the Markov assumptions:

$$\mathrm{p}(\mathrm{x}_{0:\mathrm{T}}, m, z_{0:T}, u_{0:T-1}) = p_{0|0}(x_0, m) \prod_{t=0}^{T} p_h(z_t|x_t, m) \prod_{t=1}^{T} p_f(x_t|x_{t-1}, u_{t-1})$$

where $p_h(z_t|x_t, m)$ is the observation model and $p_f(x_t|x_{t-1}, u_{t-1})$ is the motion model.

To implement the SLAM approaches, appropriate mapping technique and filter model need to be chosen. In this report particle filter and lidar-based occupancy grid map is used.

Particle filter is described to use a delta-mixture to represent the pdf of the robot state at time t:

$$\mathrm{p}_{\mathrm{t|t}}(x_t) = p(\mathrm{x_t}|z_{0:t}, u_{0:t-1}) \approx \sum_{k=1}^{N_{t|t}} \alpha_{t|t}^{(k)} \delta\left(x_t; \mu_{t|t}^{(k)}\right) \qquad \mu_{t|t}^{(k)} \in SE(2)$$

Occupancy grid map is a grid **m** with free ($\mathrm{m_i} = 0$) and occupied ($\mathrm{m_i} = 1$) cells.

The solution to specified SLAM problem can be divided into following steps:

1. Initialization: Initialize particle set $\mu_{0|0}^{(k)} = \mathbf{0}$ and $\alpha_{0|0}^{(k)} = \frac{1}{N_{0|0}}$ for $\mathrm{k} = 1, \ldots, \mathrm{N}_{0|0}$.Thus the initialized pdf becomes $p_{0|0} = \sum_{k=1}^{N_{0|0}} \alpha_{0|0}^{(k)} \delta\left(x_t; \mu_{0|0}^{(k)}\right)$
   Also the map is initialized using the lidar scan at $\mathrm{t} = 0$.
2. Prediction: The mixture pdf is approximated by drawing samples directly from particles with motion model keeping track of state of each particle

$$\mathrm{p}_{t+1|t}(x) = \int p_f(x|s,u_t) \sum_{k=1}^{N_{t|t}} \alpha_{t|t}^{(k)} \delta\left(s;\mu_{t|t}^{(k)}\right) ds \approx \sum_{k=1}^{N_{t+1|t}} \alpha_{t+1|t}^{(k)} \delta\left(x;\mu_{t+1|t}^{(k)}\right)$$

where $\mu_{t+1|t}^{(k)} = f(\mu_{t|t}^{(k)}, u_t)$ is drawn with respect of its weight and $\alpha_{t+1|t}^{(k)}=\mathrm{p}_{t+1|t}\left(\mu_{t+1|t}^{(k)}\right)$

3. Update: The delta mixture pdf is evaluated through Bayes rule

$$\mathrm{p}_{t+1|t+1}(x) = \sum_{k=1}^{N_{t+1|t}} \left[\frac{\alpha_{t+1|t}^{(k)} p_h\left(z_{t+1}\middle|\mu_{t+1|t}^{(k)}, m\right)}{\sum_{j=1}^{N_{t+1|t}} \alpha_{t+1|t}^{(j)} p_h\left(z_{t+1}\middle|\mu_{t+1|t}^{(j)}, m\right)} \right] \delta\left(x;\mu_{t+1|t}^{(k)}\right)$$

Laser correlation model $\mathrm{p_h}\left(z_{t+1}\middle|\mu_{t+1|t}^{(k)}, m\right) \propto \exp\left(\boldsymbol{corr}\left(y_{t+1}^{(k)}, m\right)\right)$ is used to update particle weight.

To prevent particle depletion, resampling need to be done when $\mathrm{N}_{eff} = \frac{1}{\sum_{k=1}^{N_{t|t}}\left(\alpha_{t|t}^{(k)}\right)^2}$ is less than a threshold.

4. Mapping: The lidar-based mapping is tracked via log-odds ratio. When new scan z_{t+1} is received, if the cell is observed free then decrease the log-odds and vice versa.
5. Repeat 2-5

3. Technical Approach

3. 1 Model specification

In this specified problem, the state of robot is defined as $\mathrm{s} = (\mathrm{x}, \mathrm{y}, \theta)^{\mathrm{T}}$ by restricting the attention on the movements in x-y plane and the rotation in the yaw plane.

For Initialization, all particle set are initialized at $\mu = (0,0,0)^{\mathrm{T}}$ and during the whole process, the number of particles is set to a constant number N so that it does not update with time. Then the first lidar scan is used to initialize the map.

For motion model, the differential-drive discrete-time model with time discretization τ is used

$$\mathrm{s}_{t+1} = f(\mathrm{s}_t, \boldsymbol{u_t} + \boldsymbol{\epsilon_t}) = \mathrm{s}_t + \tau \begin{pmatrix} v_t \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \cos\left(\theta_t + \frac{\omega_t \tau}{2}\right) \\ v_t \mathbf{sinc}\left(\frac{\omega_t \tau}{2}\right) \sin\left(\theta_t + \frac{\omega_t \tau}{2}\right) \\ \omega_t \end{pmatrix}$$

Where $\mathbf{u_t} = (v_t, \omega_t)$ representing the linearly velocity and angular velocity respectively which is obtained from encoder and IMU. $\boldsymbol{\epsilon_t}$ is the 2-D Gaussian noise accounting for the actual noise of IMU sensors.

The update step is divided into three steps. scan z_{t+1} is first transformed from lidar frame to body frame, which in this case is just an offset along x-axis, and then from body frame to the world frame using transformation matrix $\mathrm{T_{b2w}} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$, where $\mathrm{R} = \begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$, and $\mathrm{p} = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix}$ is the state of $\mu_{t+1|t}^{(k)}$.

Second step is to use laser correlation model to update the weight. Laser correlation model find all points y in the grid that correspond to the scan and compute $\mathbf{corr}(\mathrm{y}, \mathrm{m}) = \sum_i 1\{m_i = y_i\}$ and normalize the weight through softmax function:

$$\mathrm{p_h}\left(z_{t+1}\middle|\mu_{t+1|t}^{(k)}, m\right) = \frac{\boldsymbol{corr}\left(y_{t+1}^{(k)}, m\right)}{\sum_v \boldsymbol{corr}\left(v_{t+1}^{(k)}, m\right)} \propto \exp\left(\boldsymbol{corr}\left(y_{t+1}^{(k)}, m\right)\right)$$

The third step is to resample the weight and particle based on stratified resampling algorithm to prevent particle depletion. This guarantees that samples with large weights appear at least once and those with small weights at most once.

For mapping, the scan of lidar is first transformed into cartesian coordinates and then transformed from lidar frame to body frame and then to the world frame. The Bresenham's line rasterization algorithm is used to obtain the occupied cells and free cells. Technically each particle should maintain its own map, but only the state of the best particle is keep tracked in terms of mapping to improve computational efficiency.

3. 2 Implementation

Before reaching to the SLAM problem, some preprocessing work need to be done. First the linear velocity of the center of gravity is computed using the data from encoder

$$\mathrm{v} = \frac{\mathrm{d_l} + d_r}{2\Delta t}, \qquad d_l = \frac{FL + RL}{2} * 0.0022, \qquad d_r = \frac{FR + RR}{2} * 0.0022$$

Then a first order low-pass filter with 8 Hz cut-off frequency is applied to the IMU data to obtain a less noisy angular velocity. Things to notice that the time stamps of encoder and lidar data is sampled around 40Hz, while the time stamps od IMU is sampled around 100Hz. So it's necessary to pick one of the time stamps as the base and coordinate other time stamps to it. This report has chosen encoder stamps as the basis and select the IMU data at closest stamps. Other temptation such as picking the closest time stamps and average over the data before and after that time stamps have been tried but it turns out the performance is not as expected.

Then a 2-D Gaussian noise is added to the control input at each time step to account for the real-world noise of the sensor, which also makes the system more robust and makes the update step possible because all particles are initialized at the origin.

The occupancy grid map is initialized using the first lidar scan by converting lidar ranges to x-y coordinate and function 'bresenham2D' is called to return the free cells. The sweeping range of lidar sensor is from -135 to 135. Since the world frame origin is set at the initial position of the robot, the first lidar scan only need to be transformed from lidar frame to the body frame, which is just an offset along x-axis with 136.73mm.

For prediction step, the linear, angular velocity and state at time **t** is inputted into the differential-drive motion model to obtain each particle's state at time $\mathbf{t+1}$. Then the transformation from body frame to world frame is obtained

$$\mathrm{x_W} = l \cdot \cos(\alpha + \theta) + x_B$$
$$\mathrm{y_W} = l \cdot \sin(\alpha + \theta) + y_B$$

Where l is the lidar range data, α is its sweeping angle, θ is the particle's rotational state, x_B and y_B are particle's pose state. Each particle maintains its own trajectory.

For update step, the function 'mapCorrelation' function is called to compute the correlation between the current lidar scan and the current map. The correlation is evaluated at a 9*9 grid cell around the particles and its representative correlation is the maximum in the small grid. Then the weight of each particle is updated through softmax function and normalization. Typically, the particle state is not updated in update step, but according to where the max correlation appears and do some shifting to the particle to make the update step more accurate.

Then the particle with the largest weight is selected as the best particle and the lidar scan is projected on the occupancy grid map according to the state of best particle. And the threshold of log-odds on the map is set to be $\pm 10\log 4$ to prevent overconfidence.

When $\mathrm{N}_{eff} = \frac{1}{\sum_{k=1}^{N_{t|t}} \left(\alpha_{t|t}^{(k)}\right)^2} \leq \frac{\mathrm{N}}{5}$, the weight is resampled through stratified resampling. The higher threshold is, the severer the resampling criterion becomes.

3. 3 Texturing

The value at position i and j of depth image can be converted into depth value, which is used for projection of the camera model. The corresponding i and j position at RGB image can also be obtained. Base on the pinhole camera model and parameters given, the transformation from pixel on depth image to world frame is obtained as :

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f_{s_u} & f_{s_\theta} & c_u \\ 0 & f_{s_v} & c_v \\ 0 & 0 & 1 \end{bmatrix} \frac{1}{Z_o} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{oc} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{cw} & p_{cw} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Where Z_o is the depth value, R_{oc} is the rotation from regular to optical frame, R_{cw} and p_{cw} are transformation obtained from the best particle state at the current time step.

According to the equation above, the x and y position in world frame is obtained. A threshold on its z value is set to filter noise and then the corresponding RGB color information is set for those cells in the world frame.

However, the texturing algorithm is not as expected in this report.

4. Results

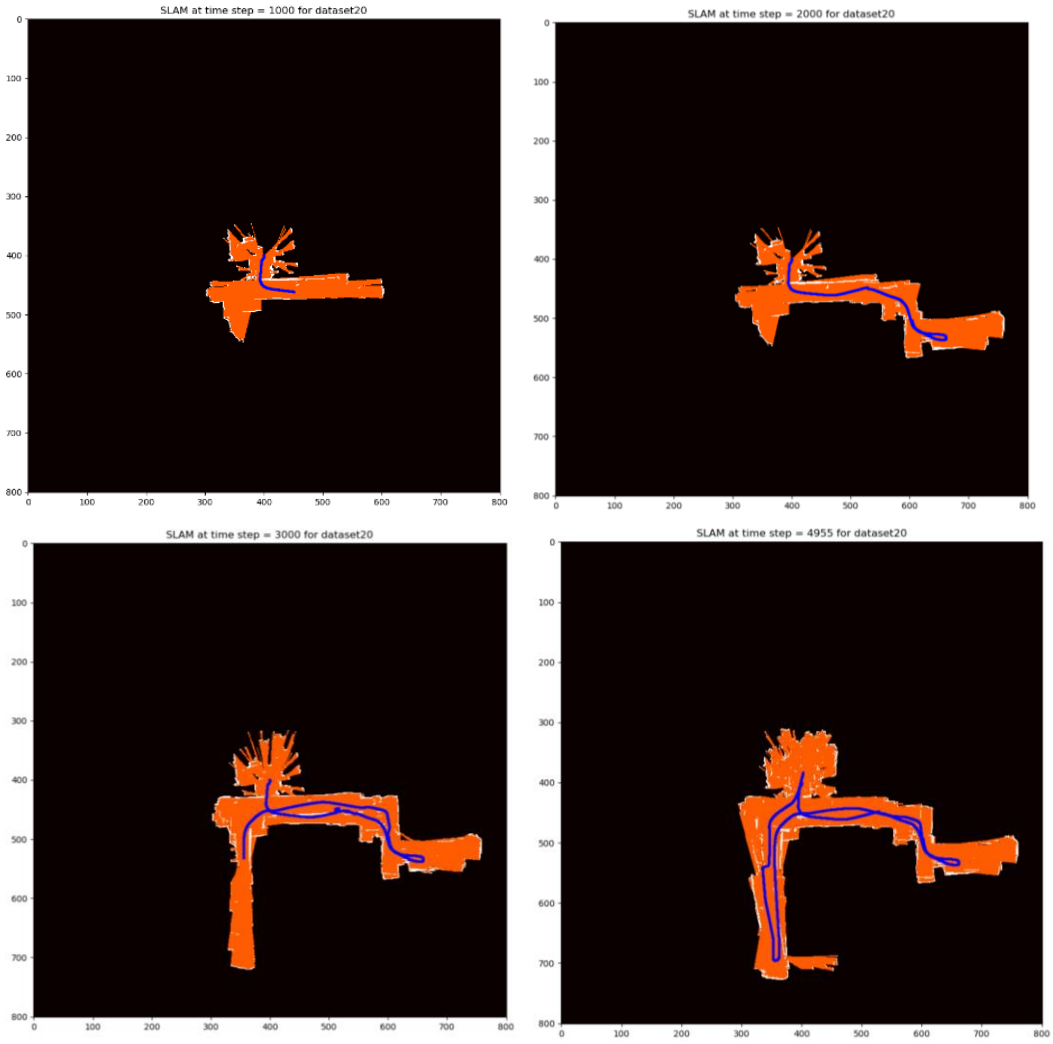

Figure 1: Mapping at different time step for data set 20

As it can be seen from results of dataset 20, there are two large shifts along the trajectory of the best particle. The mapping is not very successful when robot turn 180 degree at the bottom and some of the free cell of lidar scan are mistakenly projected into wrong places. The turning around problem is quite tricky since the IMU data becomes very sensitive to the time steps during turning around motion so any larger noise could result in the wrong state of the best particle . The approach to reduce the error is to increase the number of particles.

Another reason might come from the fact the map correlation function returns multiply best particle in terms of correlation, so the particle was randomly picked that might not be the best particle. Also when the particle's state is corrected using the index of the largest correlation of the best particle, there might also be multiply max location so a random shift is committed instead of the desired correction direction.

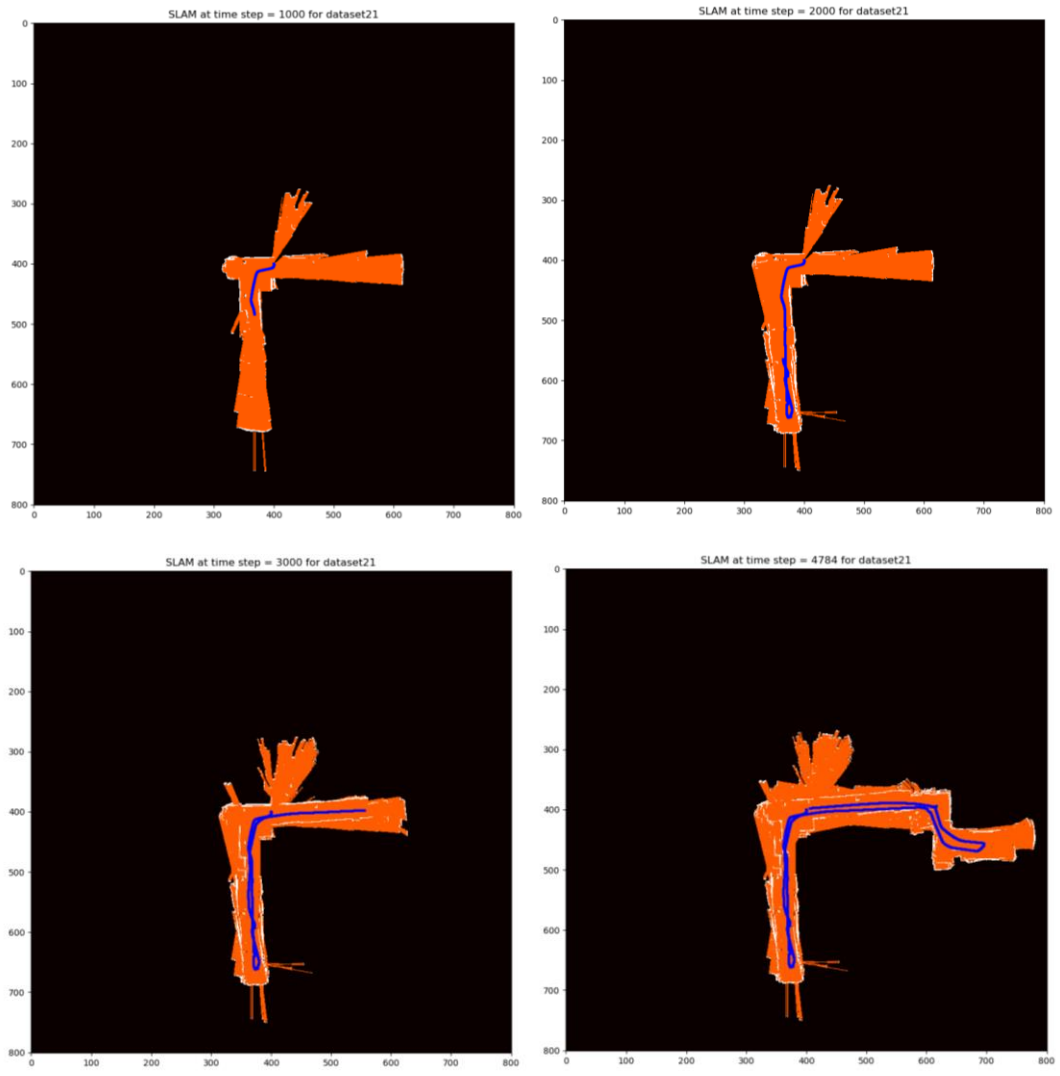

Figure 2: Mapping at different time step for dataset 21

Same phenomenon happens in dataset 21. During the forward motion, the mapping is quite normal and lidar scan is accurately projected onto the map. However, when robot turns around, some of the lidar scan is projected outside the wall, meaning the state of x or y is wrong at that time step.

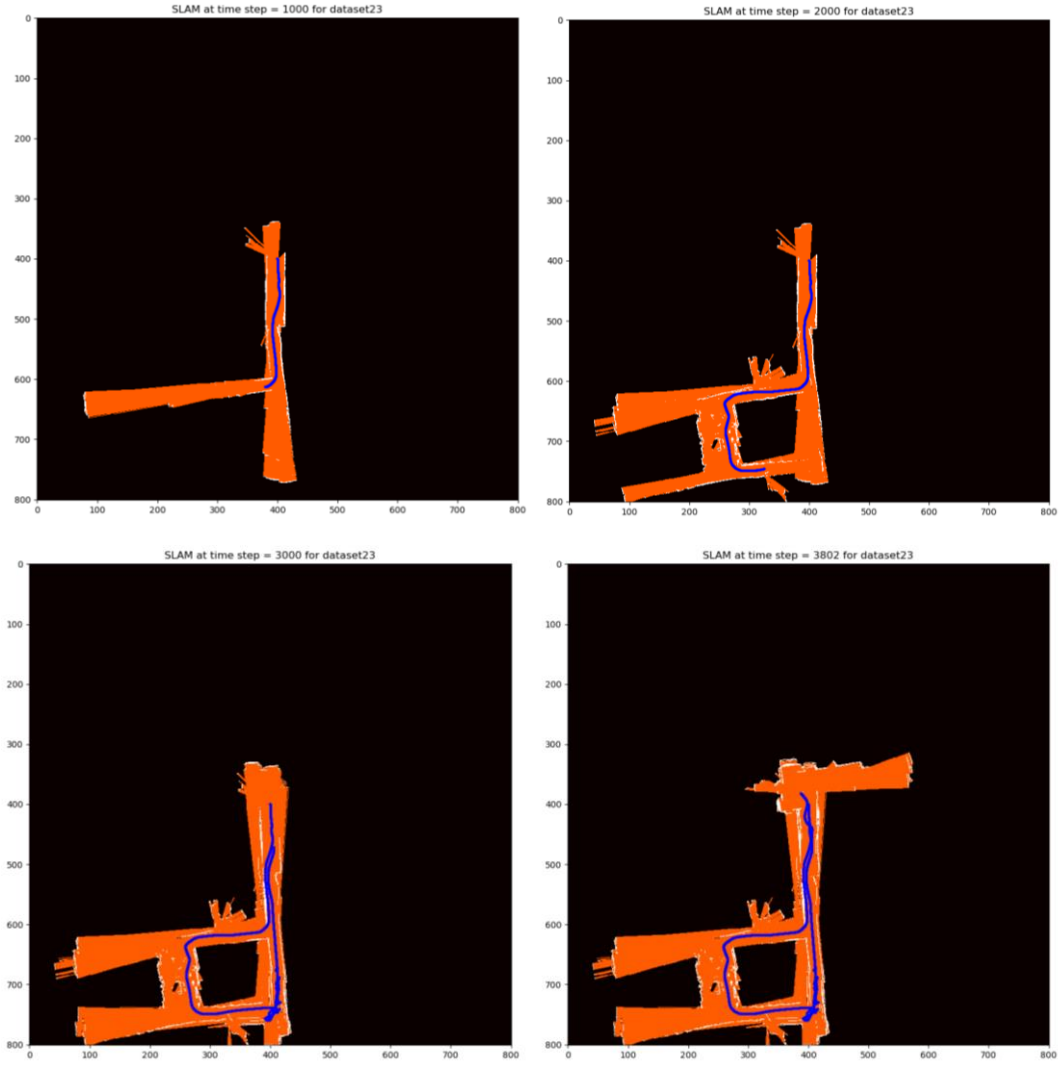

Figure 3: Mapping at different time step for dataset 23

For test set, the condition is better since there is only one turn around motion. From the simulation results we can see that the solution in this report is not able to handle motion like turn around or spinning. The main reason of this phenomenon is that map correlation is not returning the best evaluate of correlation or the noise is not tuned to an acceptable range.

The textured mapping of dataset 20 and 21 are shown as following. As it can be seen, the result is not very successful. Possibly reason for this result might be that the threshold on z-axis is not set up appropriately or the trajectory of the best particle is not representing the real condition. Also it turns out that this texturing algorithm is not very good at dealing with turn around motion.

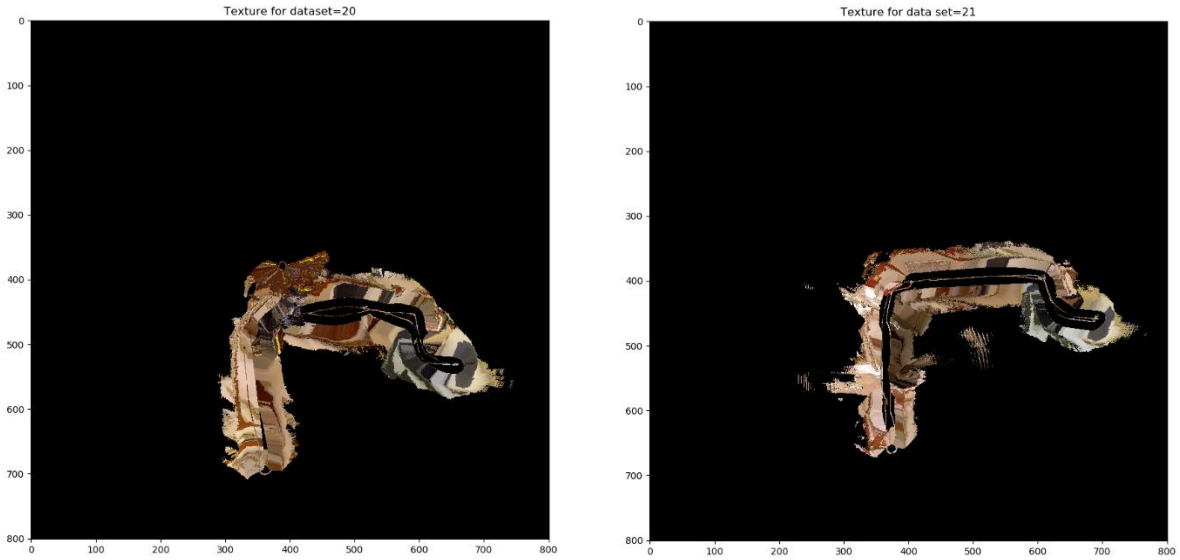

Figure 4: Textured map of dataset 20 and 21 respectively