A project report on

"EFFECT OF SWITCH VARIABLES ON ROBUST LOOP CLOSURES"

By

Arka Sadhu

Under the guidance of

Prof. Juho Kannala

Department Of Computer Science

Aalto University

Espoo, Finland

May'16-Jul'16

INTRODUCTION

Three Dimensional reconstruction of point clouds from images is a very popular topic in computer vision. But pure image based reconstruction is far from perfect especially in indoor environments. This is because indoor environments usually lack distinct textures (most corridors look the same in a particular indoor environment). This results in false loop closures.

Fortunately pure image based reconstruction can be supplemented by raw odometry data, which give information about the pose of the camera (rotation and translation with respect to the initial camera) and it may be possible to resolve some ambiguous cases and discard the false loop closures. The raw odometry data is usually not very accurate and as such there is always drift involved because of inaccuracy of the motion sensors. For small environment the drift is not noticeable but for a bigger indoor environment the drift is noticeable because of cumulative errors. To correct this we use the visual data. If we are able to identify the loop closures correctly we can then correct the point cloud, and therefore reduce the error due to drift.

The proposed method [1] for 3D scene reconstruction is based on image sequence captured by monocular cameras and the corresponding odometry information. The experiment uses Google Tango Device which provides both depth maps and color images (RGB-D data) and relatively good and robust odometry information.Using a global optimization, we adjust the relative poses so as to get a point cloud with minimum error.

The concept can be used to recreate better models of Indoor Maps, which will let us spend less time in searching the buildings floor by floor, and be a very useful navigation system.

METHODS

The images and poses data are captured using the Google Tango Device. The VTK files and JPG files are read into the program for post processing. There are several steps in this. First we extract the raw odometry data, and using the camera parameters and poses collected from the tango device, we triangulate the points to 3D world coordinates using the open source 3D vision library OpenMVG. Then we cluster the images which have visual correspondences using connected component clustering. Each cluster is fed into SfM for view clusters. For the SfM, only RGB data is used, because the input depth maps are sparse and majority of pixels lack depth. Since the view clusters may have different scales we compute the scale factor using least square method, which removes the ambiguity associated with SfM.

Then comes the Global Optimization approach.There are 5 error terms that we are considering. The first is the 2D re-projection error $E_{2D,f,j,c}$ which is the squared distance between a 3D point and x_f^j's projection $\hat{n}$ onto the camera and the observed image coordinates n_c^f.

$$E_{2D,f,j,c} = ||n_c^f - \hat{n}||^2$$

The second error term is the 3D alignment term $E_{3D,f,m}$, which is basically the euclidean distance between two 3D points which are supposed to represent the same point in space.

$$E_{3D,f,m} = ||x_f^k - x_f^l||^2$$

The third and the fourth error term is the error in relative poses between sequential images using current pose images.

$$E_{\Delta R,i} = ||\tau(\tau(R_{i-1}, R_i), \tau(\hat{R_{i-1}}, \hat{R}_i))||^2$$

$$E_{\Delta t,i} = ||\gamma(t_{i-1}, t_i) - \gamma(\hat{t_{i-1}}, \hat{t}_i))||^2$$

where

$$\tau(R_l, R_j) = R_j * R_l^T$$

$$\gamma(t_l, t_j) = t_j - \tau(R_l, R_j)\, t_l$$

The final global optimization formula is given by

$$\underset{\mathbf{x},\mathbf{r},\mathbf{t},w}{\arg\min} \sum_{f\in F} \sum_{j=1}^{M} \sum_{c\in C_f^j} \frac{E_{2D,f,j,c}}{\sigma_p^2} + \sum_{p=2}^{T} \frac{E_{\Delta R,p}}{\sigma_r^2} + \sum_{p=2}^{T} \frac{E_{\Delta t,p}}{\sigma_t^2} + \sum_{f\in F} \sum_{m\in\varphi_f} \frac{w_m \cdot E_{3D,f,m}}{\sigma_a^2} + \sum_{f\in F} \sum_{m\in\varphi_f} \frac{||1-w_m||^2}{\sigma_c^2}$$

Here **r** denotes the angle axis representation of rotations **R**. Also since different errors have different units they are weighed with inverse of corresponding measured variances. We note the introduction of the variable w which acts as a switch variable [2] which helps in reducing the computation time. Basically w is the weight given to the error. The last error term is just the correction term associated with it, because $w = 1$ is the ideal value.

In the experiment we have captured the different floors of TUAS and CS building in the Aalto University using the Google Tango device. The code has been run with varying parameters of σ_a^2 and σ_c^2.

DISCUSSION

We note the differences for the different values of σ_a^2 and σ_c^2. In [2] the empirical value is found to be around 1. But this doesn't work for every case. For the given dataset it is found
Good results : Figure (4) (7) (8)
Bad results : Figure (2) (3) (5) (6)

As can be noted from the results are much better when both the $\sigma_a and \sigma_c$ values are comparable. This is kind of expected since one of the terms is supposed to cancel the effect of other, so both should be of similar magnitudes.

There are quite a few shortcomings with using the default values for σ_a^2 and σ_c^2 as 1 as can be seen in Figure (3). The initial railing is not being shown in the point cloud. This problem also exists in (2), (5), and (6). The generated point cloud is much better for smaller values of σ_a^2 and σ_c^2. This contrast can be especially seen between Figure (2) and (4) where the former has distorted a linear path. The same problem exists with (5) and (6).

Thus we can safely conclude that we need to have similar values for both σ_a^2 and σ_c^2 and also they should be in the order of 10^{-4} to 10^{-6}. That is using correct values of σ_a^2 and σ_c^2 gives us a non-distorted path.

Figure 1: Start of service

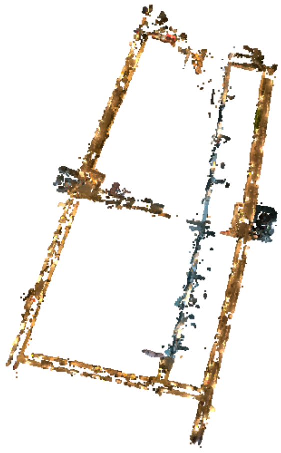

Figure 2: $\sigma_a^2 = 0.1$ $\sigma_c^2 = 0.1$

Figure 3: $\sigma_a^2 = 1$ $\sigma_c^2 = 1$

Figure 4: $\sigma_a^2 = 0.0001$ $\sigma_c^2 = 0.0001$

Figure 5: $\sigma_a^2 = 0.0001$ $\sigma_c^2 = 1$

Figure 6: $\sigma_a^2 = 1$ $\sigma_c^2 = 0.0001$

Figure 7: $\sigma_a^2 = 0.000001$ $\sigma_c^2 = 0.0001$

Figure 8: $\sigma_a^2 = 0.000001$ $\sigma_c^2 = 0.000001$

Bibliography

[1] Zakaria Laskar, Sami Huttunen, Daniel Herrera C., Esa Rahtu and Juho Kannala. *Robust Loop Closures for scene Reconstruction by combining Odometry and Visual Correspondences*

[2] Niko Sunderhauf and Peter Protzel *Switchable Constraints for Robust Pose Graph SLAM*