Recursive Total Variation Filtering Based 3D Fusion

M. A. A. Rajput[1], E. Funk[1], A. Börner[1] and O. Hellwich[2]
[1]*Department of Information Processing for Optical Systems, Institute of Optical Sensor Systems, German Aerospace Center, Berlin, Germany*
[2]*Computer Vision and Remote Sensing, Technical University Berlin, Berlin, Germany*

Keywords: Large Scale Automated 3D Modelling, Mobile Robotics, Efficient Data Structures, 3D Database.

Abstract: 3D reconstruction from mobile image sensors is crucial for many offline-inspection and online robotic application. While several techniques are known today to deliver high accuracy 3D models from images via offline-processing, 3D reconstruction in real-time remains a major goal still to achieve. This work focuses on incremental 3D modeling from error prone depth image data, since standard 3D fusion techniques are tailored on accurate depth data from active sensors such as the Kinect. Imprecise depth data is usually provided by stereo camera sensors or *simultaneous localization and mapping* (SLAM) techniques. This work proposes an incremental extension of the total variation (TV) filtering technique, which is shown to reduce the errors of the reconstructed 3D model by up to 77% compared to state of the art techniques.

1 INTRODUCTION

Mobile robotic applications gained strong impetus during last years. Inspection via small vehicles, drones, underwater robots received strong interest enabling to access areas which are not accessible by humans. Before sending a robot for an observation mission, this is planned a priori. In the case of drone flights, the way points are selected on a map which are later targeted by the drone during the flight. However, when unforeseen physical effects such as wind or insufficient light conditions arise, the quality of the collected data drops. Only after a mission has been accomplished, the user can react on the missing scans and plan a new mission. This however, is cumbersome in most cases as the light, weather or access permissions change.

For this and similar reasons mobile robotics community focuses on the research and development of real time 3D reconstruction and modeling. This would make it possible to validate the data acquired by a robot in real time and to correct the mission accordingly. Unfortunately, 3D reconstruction is a notoriously difficult topic which generally suffers from low quality of the 3D reconstruction even if the computations take hours or even weeks. This issue is even intensified when the 3D reconstruction and modeling has to run in real time, which naturally further decreases the quality of the obtained 3D data.

However, building a system that can perform incremental real-time dense large-scale reconstruction is crucial for many applications such as robot navigation (Dahlkamp et al., 2006; Christopher Urmson et. al, 2008) wearable and/or assistive technology (goo, 2014; Hicks et al., 2013), and change detection (Taneja et al., 2013). Recently a few methods have been proposed to perform incremental reconstruction (fusion) of the observed geometries from RGB-D measurements (Funk and Börner, 2016; Kähler et al., 2015; Steinbruecker et al., 2014).

These however, do not deal with the noise in the data which prohibits to apply the methods on RGB-D images from (stereo) camera sensors. This motivated the research on an incremental noise suppression technique, which is presented in this work. The presented approach relies on total variation (TV) filtering, which is known to perform well when applied in offline-processing. The main contribution of this work is the extension of the time consuming TV computations to a recursive form, enabling to re-use the previously computed model and thus to enable robust incremental 3D reconstruction in real time systems.

The underlying work presents a voxel-based incremental 3D fusion approach which makes it possible to create and to store large 3D models and maps. Section 2 introduces details from the state of the art research on incremental 3D modeling, aligned to the proposed approach. Section 3 states the research objectives investigated in this work. In Section 4.2

Rajput, M., Funk, E., Börner, A. and Hellwich, O.
Recursive Total Variation Filtering Based 3D Fusion.
DOI: 10.5220/0005967700720080
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 5: SIGMAP*, pages 72-80
ISBN: 978-989-758-196-0

the proposed method is presented, which clarifies the novelty of this work. Section 5 demonstrates the application of the technique on a benchmark dataset with an available ground truth leading to quantitative comparison between the proposed and two selected techniques. Finally, concluding remarks and aspects for further research are given in Section 7.

2 LITERATURE OVERVIEW

2.1 Reconstruction

Early real-time dense approaches (Stühmer et al., 2010; Newcombe et al., 2011) were able to estimate the image depth from monocular input, but their use of a dense voxel grid limited reconstruction to small volumes and to powerful GPU devices due to high memory and computation requirements. Also suffering from the scalability issue, KinectFusion (Izadi et al., 2011; Funk and Börner, 2016) directly sensed depth using active sensors and fused the high quality depth measurements over time to surfaces. The scalability issue has been resolved by voxel hierarchy approaches (Chen et al., 2013) or direct voxel block hashing (Kähler et al., 2015) to avoid storing unnecessary data for free space allowing scaling to unbounded scenes. At the core, the mentioned approaches rely on the volumetric fusion arithmetic from (Curless and Levoy, 1996), which expects accurate depth measurements. Thus, the issues with error prone data remains uncovered.

2.2 Noise Suppression and Fusion

Regarding the noise suppression of error prone depth measurements, direct depth image filtering became state of the art and has been applied in several projects (Stühmer et al., 2010; Newcombe et al., 2011). Basically, the TVL_1 technique from (Rudin et al., 1992) is applied enforcing the photo consistency and depth smoothness between multiple images observing the same object. This approach delivers high quality results given depth measurements of low quality. However, the TVL_1 fusion computation is performed every time a new depth image is added to the scene leading to slow frame rates.

3 RESEARCH OBJECTIVES

This work addresses both critical challenges identified from the literature, which are: i) The reconstruction of unbounded environments. This is addressed by the method of (Funk and Börner, 2016). ii) The 3D fusion of error prone depth data into a volumetric 3D space, which is the main aspect of this work. The goal of this research was to perform TV fusion directly on the 3D voxel space while replacing the L_1 cost by a recursive L_2 optimization.

4 METHODOLOGY

4.1 Camera Model and World Representation

In order to generalize the fusion framework, we designed our system to work with series of depth images captured by pinhole based depth camera (such as Kinect camera used in KinFu (Izadi et al., 2011)) and color images captured from standard pinhole color camera. We assume that scaled focal lengths (f_x, f_y) and a central point $c = (c_x, c_y)$ of both cameras are known. At every time stamp t both cameras record respective depth image $\mathbf{Z}_t$ and a color image $\mathbf{I}_t$ along with translation $T_t \in \mathbb{R}^3$ and orientation $R_t \in SO(3)$ is also acquired. A pixel $(x,y)^T$ on image plane can be related to a global 3D point $P_w \in \mathbb{R}^3$ by

$$P_w = R_t \cdot \begin{bmatrix} (x-c_x)\frac{\mathbf{Z}t(x,y)}{f_x} \\ (y-c_y)\frac{\mathbf{Z}t(x,y)}{f_y} \\ \mathbf{Z}t(x,y) \end{bmatrix} + T_t. \quad (1)$$

P_w represents a 3D coordinate in world space. Depending upon the desired scale of reconstruction P_w can be scaled accordingly. In actual fusion stage the model can also be re-sampled in 3D voxel space, similar to world coordinate system each voxel can be accessed as $V_{(x,y,z)} \in \mathbb{R}^3$.

In volumetric depth fusion approaches (Curless and Levoy, 1996; Kähler et al., 2015; Steinbruecker et al., 2014), a signed distance function (SDF) is computed for all voxels which are in near proximity of the implicit surface. However our proposed scheme uses only those voxels which lie along-the-ray from camera center Cam_w and a 3D sample position in world space P_w, this selection of voxels referred as SDF-signal as shown in figure 1. In our implementation we use linear Truncated Signed Distance Function (TSDF) which ranges between $[+1,...,-1]$, which results in implicit representation having zero-crossing at the given depth. These SDF-Signals are then used directly in recursive fusion framework.

Since the main objective of proposed scheme is to optimize depth fusion and enhance overall 3D reconstruction, we incorporate ground truth camera poses during fusion process. This assumption prevents all

the problems related to failed tracking in SLAM and ensures that optimization is focused on depth fusion.

4.2 Recursive TV 3D Fusion

Proposed scheme uses the recursive aspect of Least Square Estimation (LSE) in which depth fusion is achieved by solving the system of linear equation. Assuming $x \in \mathbb{R}^n$ is SDF signal containing current state of system and $y \in \mathbb{R}^n$ is SDF signal generated from particular depth sample with n being the ray support length in voxels. In order to fuse these two SDF signals using least squares we assume that transformation function between x and y is linear in nature. Thus, least square estimate $\hat{x}$ is that value of y that minimizes the residual error

$$\|Ax+y\|_2^2 \tag{2}$$

In order to add inherent Total Variation filtering based regularization in proposed technique, we can take advantage by accessing the values in neighboring voxels. Hence minimization function for such regularized recursive system can be re-written as

$$\|Ax+y\|_2^2+\lambda\|g(x)\|_2^2 \tag{3}$$

Where λ is regularization parameter and $g(x)$ is function which approximate the second order finite difference. Although the number of voxels in SDF signal (referred compactly as *Support* in further description) depends upon the scale of reconstruction and noise, for explanation purpose we assume support $n = 5$ voxels. Assuming that $v_{k+1} = \{a_1, a_2, a_3, ...a_n\}$ is a SDF signal represented as a vector containing all the voxels on a ray through a 3D sample (Figure 1) and a_i represents the SDF values of these voxels before the update the Equation 3 can be written as

$$\|Ax+y\|_2^2+\lambda\|Dv+C\|_2^2 \tag{4}$$

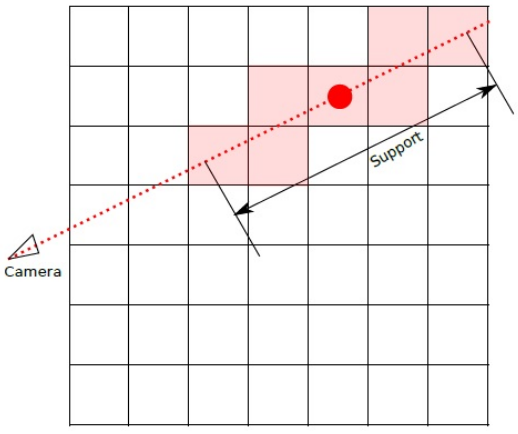

Figure 1: SDF signal and selected voxels.

Actual derivation of D, C matrices and simplification of Equation 4 is beyond the scope of this paper however detail steps can be found in the Appendix 7. Next, the proposed technique is compactly referred to as *RFusion*.

4.3 Update Algorithm

Algorithm 1 shows a simplified version of RFusion which describes how the SDF signal is accessed and updated. In actual implementation the algorithm was implemented using threads to maximize the efficiency and utilize the latest CPU architecture. We start our fusion application by instantiating an object of data structure proposed by (Funk and Börner, 2016) in line 1. For each $pixel(row, col)$ individual steps for fusion algorithm are:

- calculation of 3D world coordinate P_w at (*row, col*), line 10
- getting RGB value at (*row, col*), line 11
- generating linear TSDF values (i.e. $[+1, ..., -1]$) depending upon the *support*, line 12
- reading current SDF value from voxels and creating SDF signal from W, line 13
- getting value of d and c matrices using method describe in Appendix 7
- compute the new SDF signal y_t from previous estimate $\hat{x}_{t-1}$, line 15
- updating the W with latest estimate for particular depth sample in line 16

4.4 Effects of Regularization

In order to investigate and visualize the effects of regularization parameter λ on noise suppression and optimized depth fusion, we implemented 2D variant of the proposed scheme. To avoid the effects of fusion from regularization, single synthetic surface is fused with itself. It was observed that TV filtering produced inherent smoothing effect on voxels at the time of SDF signal fusion. This regularization effect is demonstrated in Figure 2, where red cells represent positive values and blue cells represent negative TSDF values.

5 EXPERIMENTS

5.1 ICL RGBD Dataset

In order to analyze the working of proposed scheme we used synthetic RGBD dataset provided by ICL-NUIM (Handa et al., 2014) which is considered to be

Algorithm 1: Simplified RFusion algorithm.

let $W \leftarrow$ DataStructure (from (Funk and Börner, 2016))
procedure FUSE
  $t \leftarrow$ currentFrame
  $\mathbf{I}_t \leftarrow$ RGB image and $\mathbf{Z}_t \leftarrow$ Depth image
  $T_t \leftarrow$ Translation
  $R_t \leftarrow$ Rotation
  support $\leftarrow 5$
  for $i = 0$ to *rows*-1 **do**
    for $j = 0$ to *cols*-1 **do**
      $P_w \leftarrow R_t . \begin{bmatrix} (x-c_x)\frac{\mathbf{Z}t(x,y)}{f_x} \\ (y-c_y)\frac{\mathbf{Z}t(x,y)}{f_y} \\ \mathbf{Z}t(x,y) \end{bmatrix} + T_t .$
      color $\leftarrow \mathbf{I}_t$.getColor(i,j)
      $y_t \leftarrow$getTSDF()
      $\hat{x}_{t-1} \leftarrow W$.readVoxels($P_w$,$T_t$,support)
      [d,c] $\leftarrow W$.getDC(x) (from eq. 20)
      $\hat{x}_t \leftarrow \hat{x}_{t-1} + P_t \left[\phi(y_t - \lambda d^T c) - (\phi^T \phi + \lambda d^T d)\hat{x}_{t-1} \right]$ (from eq. 13)
      W.updateSystem($\hat{x}_t$,color)
    end for
  end for
  $t \leftarrow t+1$
end procedure

standard in depth fusion and 3D reconstruction evaluation. We have tested proposed scheme with and without noise for thorough evaluation.

5.2 Evaluation

Since the dataset is synthetic in nature, it is possible to evaluate reconstructed model with actual model afterwards. After necessary alignment and scaling, for each vertex in the reconstructed model a closest triangle is registered and perpendicular distance is recorded. Five standard statistics (Mean, Median, Standard Deviation, Min and Max etc) can be computed from the recorded distance as suggested by (Handa et al., 2014).

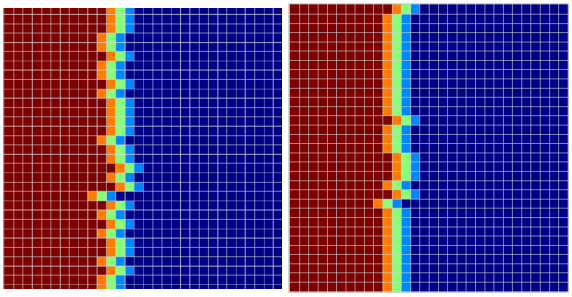

Figure 2: (left) illustrates an implicit surface with noise; (right) demonstrates the effects of regularization parameter.

However we will analyze the performance of each framework with mean and standard deviation of standard statistics.

Candidates for quantitative comparison are following fusion frameworks:

1. Fast Fusion (Steinbruecker et al., 2014)
2. RFusion
3. InfiniTAM [1] (Kähler et al., 2015)

It is worth mentioning that similar to proposed technique, ground truth camera poses ware used instead of tracking to avoid any biased evaluation. Figure 3 illustrates the reconstruction with our proposed technique vs the reconstruction from FastFusion. It was observed that RFusion was able to fuse minute details of model (such as lamp pole, leaves of plant etc) into single connected mesh however FastFusion generated multiple inconsistent meshes.

Table 1: Comparision of absolute surface error (in meters).

Dataset \ Method	RFusion	FastFusion	InfiniTAM
LR0	**0.003045**	0.011895	0.008900
LR1	0.002947	0.011204	**0.002900**
LR2	**0.003183**	0.006634	0.008800
LR3	**0.002978**	0.018180	0.041000

During experimentation phase it was observed that RFusion gives improved performance compared to state of art fusion techniques and achieved 4-5 frames per second on pure CPU based implementation.

System used for experimentation has the following specifications: Intel Core i7-4790, 8GB RAM with Windows 7 (64-bit) OS.

Table 2: Comparision of absolute surface error on noisy dataset (in meters).

Dataset \ Method	RFusion ($\lambda = 0.3$)	FastFusion
LR0	**0.01336**	0.05672
LR1	**0.01416**	0.07523
LR2	**0.01979**	0.07082
LR3	**0.02090**	0.06643

[1] Since we ware not able to modify the working of InfiniTAM with ICL-NUIM dataset we are using the Mean values published in (Kähler et al., 2015) for this dataset

Figure 3: Screenshots from reconstructed model with RFusion (upper row) and FastFusion (bottom row).

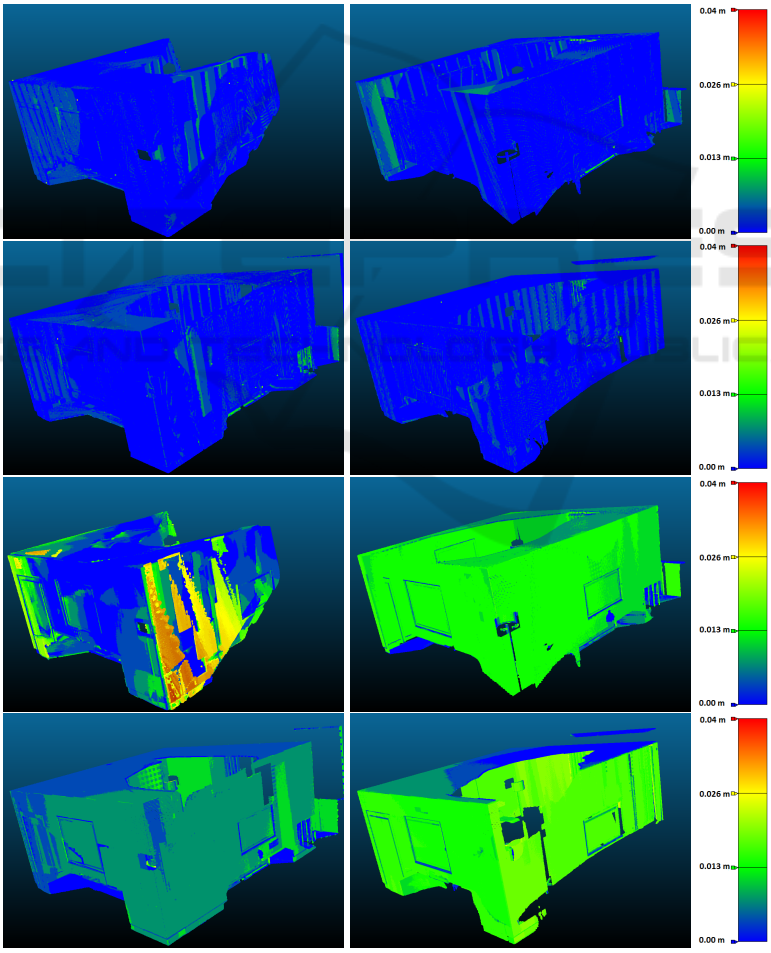

Figure 4: Color coded errormaps from RFusion (Upper 2 rows) vs FastFusion (Bottom 2 rows) along with absolute color scale used to genarate errormaps.

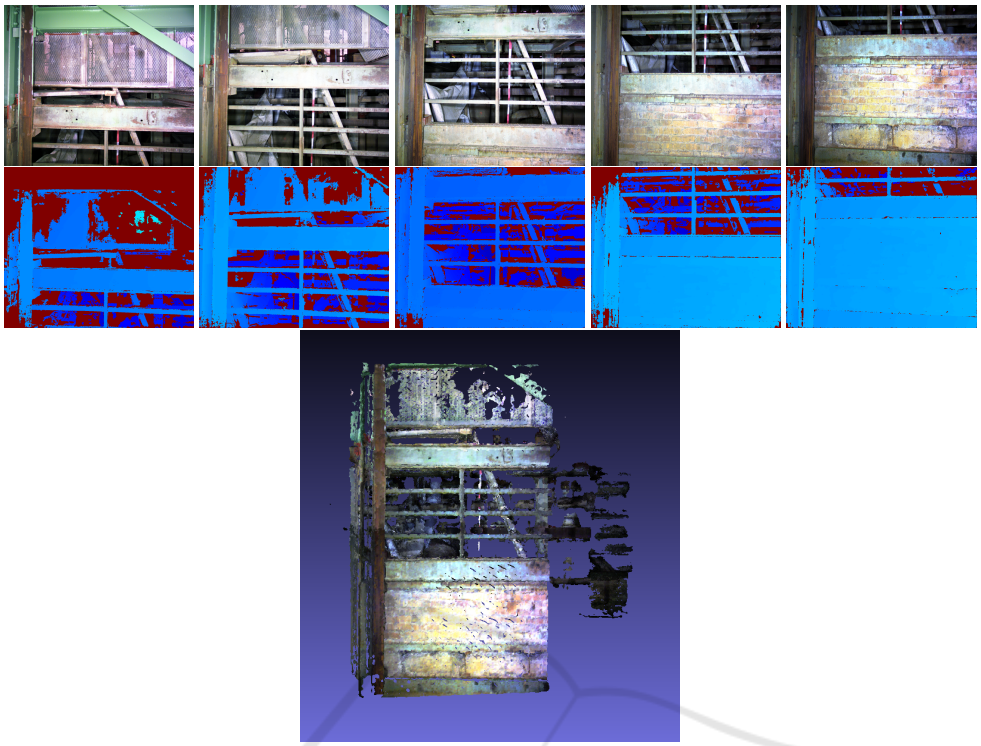

Figure 5: RGB images (upper row), depth images (second row) and reconstructed 3D model (last row).

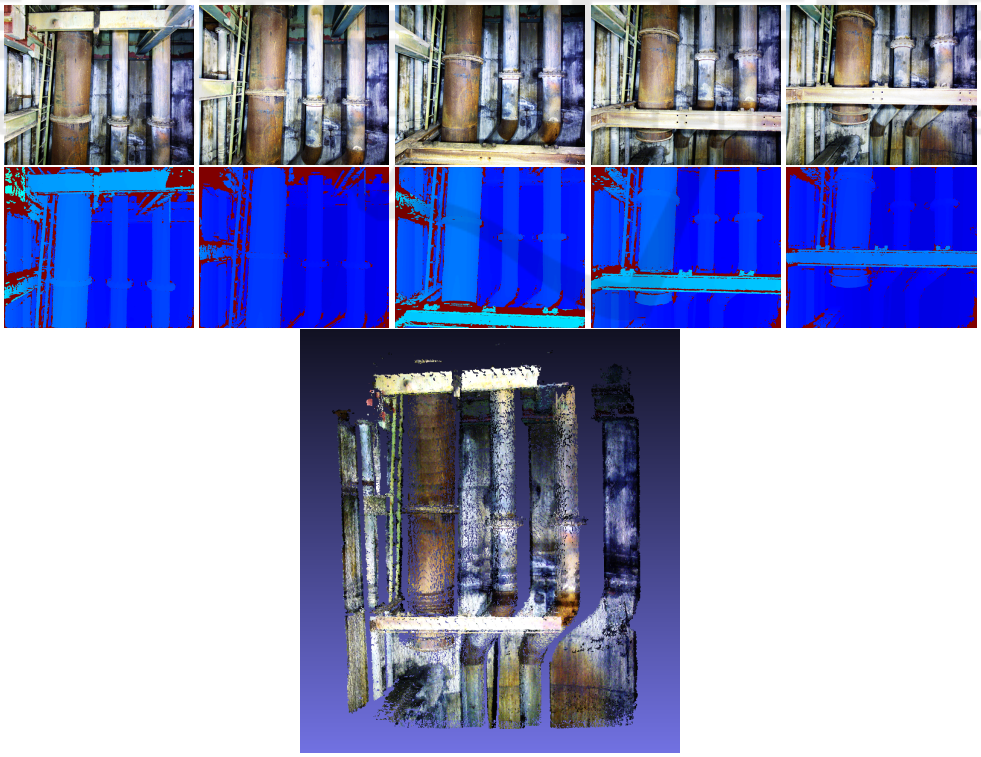

Figure 6: RGB images (upper row), depth images (second row) and reconstructed 3D model (last row).

6 EXPERIMENTATION RESULTS FROM IPS DATA

In order to evaluate the working of proposed system with real data, we used dataset captured from stereo camera based IPS (Baumbach and Zuev, 2014). Since the data is real in nature, it is not feasable to evaluate the absolute surface error of reconstructed model and actual environment. Therefore Screenshots of such reconstruction are illustrated in figure 5 and 6 for visual inspection and evaluation.

7 CONCLUSION AND OUTLOOK

In this paper, we presented a novel approach to address the challenges related to 3D depth fusion and reconstruction with the use of L2 regularization based recursive fusion framework. We demonstrated that the proposed system has potential of reducing noise along with capability of incremental 3D depth fusion. At current state, implementation of proposed scheme is purely threads based CPU processing, however further implementation is required to extend the framework to utilize latest GPU computation power along with CPU processing. Furthermore, since the system handles noise inherently, it would be interesting to integrate planar simplification techniques for improved 3D reconstruction in future research exploration.

REFERENCES

(2014). ATAP Project Tango Googl. http://www.google.com/atap/projecttango/. Accessed: 2015-11-22.

Baumbach, D. G. D. and Zuev, S. (2014). Stereo-Vision-Aided Inertial Navigation for Unknown Indoor and Outdoor Environments. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2014* . IEEE.

Chen, J., Bautembach, D., and Izadi, S. (2013). Scalable real-time volumetric surface reconstruction. *ACM Trans. Graph.*, 32(4):113:1–113:16.

Christopher Urmson et. al (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):425–466.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA. ACM.

Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., and Bradski, G. (2006). Self-supervised monocular road detection in desert terrain. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA.

Funk, E. and Börner, A. (2016). Infinite 3d modelling volumes. In *VISAPP 2016*.

Handa, A., Whelan, T., McDonald, J., and Davison, A. J. (2014). A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1524–1531. IEEE.

Hicks, S. L., Wilson, I., Muhammed, L., Worsfold, J., Downes, S. M., and Kennard, C. (2013). A depth-based head-mounted visual display to aid navigation in partially sighted individuals. *PLoS ONE*, 8(7):e67695.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. (2011). Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Symposium on User Interface Software and Technology*. ACM.

Kähler, O., Prisacariu, V. A., Ren, C. Y., Sun, X., Torr, P. H. S., and Murray, D. W. (2015). Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Transactions on Visualization and Computer Graphics (Proceedings International Symposium on Mixed and Augmented Reality 2015*, 22(11).

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). Dtam: Dense tracking and mapping in real-time. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2320–2327, Washington, DC, USA. IEEE Computer Society.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(14):259 – 268.

Steinbruecker, F., Sturm, J., and Cremers, D. (2014). Volumetric 3d mapping in real-time on a cpu. In *Int. Conf. on Robotics and Automation*, Hongkong, China.

Stühmer, J., Gumhold, S., and Cremers, D. (2010). Real-time dense geometry from a handheld camera. In *Pattern Recognition (Proc. DAGM)*, pages 11–20, Darmstadt, Germany.

Taneja, A., Ballan, L., and Pollefeys, M. (2013). City-scale change detection in cadastral 3d models using images. In *Computer Vision and Pattern Recognition (CVPR)*, Portland.

APPENDIX

Derivation of RFusion

In this section we derive the equations of RFusion, for the sake of better readability we will simplify the equations for 2D fusion system rather than 3D fusion system. Assuming n = support of SDF signal, $\hat{x}$ is estimated state of system for particular 3D voxel and y = new TSDF signal. Then such system can easily be

describe by the following equation.

$$y = \phi\hat{x}$$
$$\hat{x} = (\phi^T\phi)\phi^T y \quad (5)$$

In order to integrate the true aspect of second order finite difference it is assumed that

$$\hat{Y} = Y - \lambda D^T C \quad (6)$$

Since equation 5 is only valid if we have single input SDF signal, we assume that we have multiple SDF signals, Then we can extend equation 5 for batch based least square system as follows

$$\begin{bmatrix} y_{0*1} \\ y_{0*2} \\ \dots \\ y_{0*n} \\ y_{1*1} \\ \dots \\ y_{m*n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \dots & & & & \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ \dots & & & & \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \hat{x}$$

where m is the number of SDF signals about to be merged

$$\hat{Y} = \Phi x$$
$$x = (\Phi^T\Phi)\Phi^T\hat{Y}$$

The regularized variant of batch-based system can be re-written as

$$\hat{x} = \left(\Phi^T\Phi + \lambda D^T D\right)\Phi^T\hat{Y} \quad (7)$$

where λ is a regularization parameter. In order to convert batch based system to recursive system we assume that Φ, Y and D follow recursive succession

$$\Phi_{k+1} = \begin{bmatrix} \Phi_K \\ \phi \end{bmatrix} \quad \hat{Y}_{k+1} = \begin{bmatrix} \hat{Y}_K \\ \hat{y} \end{bmatrix} \quad and D_{k+1} = \begin{bmatrix} D_k \\ d \end{bmatrix}$$

Equation 7 for $k+1$ instance can be written as

$$\hat{x_{k+1}} = \left(\Phi_{k+1}^T\Phi_{k+1} + \lambda D_{k+1}^T D_{k+1}\right)^{-1}\phi_{k+1}^T\hat{Y}_{k+1} \quad (8)$$

For better readability we use substitution of P_{k+1}^{-1} from equation (15) and using the recursive versions of Φ and Y we get

$$\hat{x_{k+1}} = P_{k+1}\Phi_{k+1}^T\hat{Y}_{k+1}P_{k+1}^{-1}\hat{x_{k+1}} \quad = \Phi_{k+1}^T\hat{Y}_{k+1} \quad (9)$$

Similarly for k^{th} instance

$$\Phi_k^T\hat{Y}_k = P_k^{-1}\hat{x_k} \quad (10)$$

Resuming from equation (9) we get

$$\hat{x_{k+1}} = P_{k+1}\left[\Phi_k^T\hat{Y}_k + \phi^T\hat{y}_{k+1}\right]$$

By using the value of $\Phi_k^T Y_k$ from equation (10) we get

$$\hat{x_{k+1}} = P_{k+1}\left[P_k^{-1}\hat{x_k} + \phi^T\hat{y}_{k+1}\right]$$

By using the value of P_k^{-1} from equation (16) we get

$$\begin{aligned}\hat{x}_{k+1} &= P_{k+1}\left[\left(P_{k+1}^{-1} - (\phi^T\phi + \lambda d^T d)\right)\hat{x_k} + \phi^T\hat{y}_{k+1}\right] \\ &= \left(P_{k+1}P_{k+1}^{-1} - P_{k+1}(\phi^T\phi + \lambda d^T d)\right)\hat{x_k} + P_{k+1}\phi^T\hat{y}_{k+1} \\ &= P_{k+1}P_{k+1}^{-1}\hat{x_k} - P_{k+1}(\phi^T\phi + \lambda d^T d)\hat{x_k} + P_{k+1}\phi^T\hat{y}_{k+1} \\ &= \hat{x_k} - P_{k+1}\left(\phi^T\phi + \lambda d^T d\right)\hat{x_k} + P_{k+1}\phi^T\hat{y}_{k+1} \\ \hat{x}_{k+1} &= \hat{x_k} + P_{k+1}\left[\phi\hat{y}_{k+1} - (\phi^T\phi + \lambda d^T d)\hat{x_k}\right] \quad (11)\end{aligned}$$

by using the assumption from equation (6) we can further assume that

$$\hat{y}_{k+1} = y_{k+1} - \lambda d^T c$$

Hence final equation (11) for RFusion will become

$$\hat{x}_{k+1} = \hat{x_k} + P_{k+1}\left[\phi(y_{k+1} - \lambda d^T c) - (\phi^T\phi + \lambda d^T d)\hat{x_k}\right] \quad (12)$$

From the fundamental regularized LSE equation

$$\hat{x_{k+1}} = \left(\Phi^T\Phi + \lambda D^T D\right)^{-1}\Phi^T y$$
$$\text{Let } P_k = \left(\Phi^T\Phi + \lambda D^T D\right)^{-1} \quad (13)$$
$$P_k^{-1} = \left(\Phi^T\Phi + \lambda D^T D\right) \quad (14)$$

For the recursive part we can extend the Φ and D matrices as follows

$$\Phi_{k+1} = \begin{bmatrix} \Phi_K \\ \phi \end{bmatrix} \quad \text{and } D_{k+1} = \begin{bmatrix} D_k \\ d \end{bmatrix}$$

Then equation (14) will become

$$P_{k+1} = \left(\begin{bmatrix} \Phi_K \\ \phi \end{bmatrix} \quad \begin{bmatrix} \Phi_K & \phi \end{bmatrix} \quad + \lambda \begin{bmatrix} D_K \\ d \end{bmatrix} \quad \begin{bmatrix} D_K & d \end{bmatrix} \quad \right)^{-1}$$

$$P_{k+1} = \left(\Phi_k^T \Phi + \phi^T \phi + \lambda D_k^T D + \lambda d^T d \right)^{-1}$$

$$P_{k+1} = \left((\Phi_k^T \Phi + \lambda D_k^T D) + \phi^T \phi + \lambda d^T d \right)^{-1}$$

$$P_{k+1} = \left(P_k^{-1} + (\phi^T \phi + \lambda d^T d) \right)^{-1} \tag{15}$$

$$P_{k+1}^{-1} = P_k^{-1} + (\phi^T \phi + \lambda d^T d)$$

$$P_k^{-1} = P_{k+1}^{-1} - (\phi^T \phi + \lambda d^T d) \tag{16}$$

$$P_{k+1} = \left(P_k^{-1} + \begin{bmatrix} \phi^T \phi & I \end{bmatrix} \quad \begin{bmatrix} I \\ \lambda d^T d \end{bmatrix} \quad \right)^{-1}$$

For simplicity assuming

$B = \begin{bmatrix} \phi^T \phi & I \end{bmatrix}$ and $C = \begin{bmatrix} I \\ \lambda d^T d \end{bmatrix}$ we get

$$P_{k+1} = \left(P_k^{-1} + BC \right)^{-1}$$

Using matrix inversion lemma

$$(A+BC)^{-1} = A^{-1} - A^{-1}B(I+CA^{-1}B)^{-1}CA^{-1}$$

$$P_{k+1} = P_k - P_k B(I+CP_kB)^{-1}CP_k \tag{17}$$

Formulation of D Matrix

Since we are dealing with elements in a 2D matrix each cell and respective neighboring cells can be accessed by their respective spatial information (i.e. row and column values in case of 2D), however actual implementation of proposed technique is carried to handle 3D depth fusion. For each voxel value a_k (where $0>k>support$) in vector SDF signal v, assuming that i and j are index values of row and column respectively for accessing a_k in equation (15), finite difference in vector form can be written as

$$\nabla a_k = \begin{bmatrix} \nabla_{xx} \\ \nabla_{yy} \\ \nabla_{xy} \\ \nabla_{yx} \end{bmatrix}$$

$$\begin{bmatrix} \nabla_{xx} \\ \nabla_{yy} \\ \nabla_{xy} \\ \nabla_{yx} \end{bmatrix} = \begin{bmatrix} a(i-1,j) - 2a(i,j) + a(i+1,j) \\ a(i,j-1) - 2a(i,j) + a(i,j+1) \\ \frac{a(i+1,j+1) - a(i+1,j) - a(i,j+1) + 2a(i,j) - a(i-1,j) - a(i,j-1) + a(i-1,j-1)}{2} \\ \frac{a(i+1,j+1) - a(i+1,j) - a(i,j+1) + 2a(i,j) - a(i-1,j) - a(i,j-1) + a(i-1,j-1)}{2} \end{bmatrix} \tag{18}$$

Elements of equation (15) can be separated depending upon if the elements are in the incident ray which is currently being fused or in neighboring cell. The separated elements can then be written using multiple matrix form as

$$\nabla a_k = D_k v + C_k \tag{19}$$

where

$$D_k = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ -2 & 0 & 0 & \dots & 0 \\ 1 & 0.5 & 0 & \dots & 0 \\ 1 & 0.5 & 0 & \dots & 0 \end{bmatrix}$$

$$C_k = \begin{bmatrix} a(i-1,j) \\ a(i,j-1) + a(i,j+1) \\ \frac{a(i+1,j+1) - a(i+1,j) - a(i-1,j) - a(i,j-1) + a(i+1,j+1)}{2} \\ \frac{a(i+1,j+1) - a(i+1,j) - a(i-1,j) - a(i,j-1) + a(i+1,j+1)}{2} \end{bmatrix}$$

D_k and C_k matrix in equation (19) are only valid[2] for a_k (where $k = 1$). However by using the same method, composite D and C matrix can be formulated and written as

$$\nabla v = \begin{bmatrix} \nabla a_1 \\ \nabla a_2 \\ \dots \\ \nabla a_n \end{bmatrix} = \begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_n \end{bmatrix} v + \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_n \end{bmatrix}$$

$$\nabla v = Dv + C \tag{20}$$

Matrix C from equation (20) is used in the later stages of RFusion to incorporate the integrated smoothing.

[2]Values of D and C matrices are calculated on run time, hence elements depend upon the angle of ray, size of SDF width etc.