Multi-View 3D reconstruction using Convex Optimization

Nimesh Khandelwal
Roll No: 20105284
Project Mode: **mixed**
nimesh20@iitk.ac.in

***Abstract*—Multi-View 3D reconstruction is an important research problem that has been explored in great depth by many research groups as an attempt to generate near accurate virtual models of real objects/environments. The current state-of-the-art system uses a technique know as bundle adjustment to correct the final 3D points of the model. Other techniques include minimizing the surface energy of the inaccurately reconstructed model. This paper presents the study of one such technique. We use a zero-level set in the distance field representation of the reconstructed model to represent the surface of our model. This paper also presents the theoretical study of the energy functional as well as the algorithm used.**

***Index Terms*—Convex Optimization, Computer Vision, 3D reconstruction**

I. Introduction

3D object/scene reconstruction using a monocular camera is one of the fundamental problems of computer vision. It consists of many sub-problems, like feature detection and mapping follwed by tracking. To reconstruct the scene we need to establish a local coordinate system, for which we need to accurate camera pose as we move the camera. Hance, we need to calculate the camera pose from the image sequence as well. This is a part of the problems addressed by SLAM. It aims to solve both these problems simultaneously. Realtime SLAM has seen great progress recently. The initial work of [6] used statistical approach (Extended Kalman Filter). The current state-of-the-art systems use the geometric approach (Bundle Adjustment). In this work the authors split the tracking and mapping tasks into two different threads. This system is called the Parallel Tracking and Mapping (PTAM). This program is now not compatible with recenet versions of OpenCV. The other sub-problem to address is the generation of depth maps from the pair of stereo images. Authors of [7] provide an accurate and robust method for this.

There are many open source libraries available for the reconstruction task as well, Eg. OpenCV SFM, OpenMVG, COLMAP, PMVS, MVE and many others. For the current study, feature detection and matching, rectification of stereo image planes, disparity map generation, and the depth maps generation was done using the in-built functions provided by the OpenCV library.

For solving the optimization problem, the recently introduced first order primal-dual algorithm has been used [1]. This has an advantage of discretely minimizing the functional without performing any form of approximation.

II. Method

A. 3D Points and Camera Pose Estimation

For the current study, the method similar to the one given in [4] has been followed. In that, the authors have used the PTAM for obtaining high quality poes estimates. In the current implementation, however, we have written a custom tracking stack using the OpenCV library. For feature detection, SIFT descriptors have been used along with FLANN based KNN matcher. That was followed by the Lowe's distance ratio test to weed out unfit matches. Using the keypoints generated by the matching algorithm, the corresponding homography matrices were generated for both the frames. The homography matrices are used to transform image planes such that they are in a binocular configuration. The advantage of using the homography transform is that now one only has to look along certain lines (Epipolar Lines) in order to match pixels in one image to corresponding pixels in the another.

B. Depthmap Generation

The rectified images are used to create the disparity map for the given stereo image pair. For this, the stereoSGBM algorithm was used from the OpenCV library. After parameter tuning, the disparity results were good enough to use them for depth map calculation. For this, the traingulation algorithm was used, that is implemented in the OpenCV reprojectimageTo3D() function. This function gives the 3D coordinates of each pixel in the disparity map given the disparity-to-depth matrix of the current frame. This disparity-to-depth matrix is formed using the intrinsic parameteres of the camera only.

As for the camera pose estimates, the current implementation uses pre-existing data for the camera pose for each image. If that is not the case, then the Perspective N Point transform can be used on the matched features to calculate the change in camera pose. This is implemented in the OpenCV solvePnP() function.

C. Depthmap Fusion

To represent the model in 3D space using the points calculated by triangulation, we use the level set approach in a volumetric grid to describe them. The surface is implicitly represented as a zero-level set of the function $u : \Omega \rightarrow [-1, 1]$.

For this, a signed distance field is constructed from the 3D points calculated via triangulation. The null iso-surface of this field represents the surface of our reconstructed model. For

every 3D point, the voxel containing the 3D point is assigned the value 0. Then for each voxel lying along the line-of-sight from camera to the point, it is assigned a positive value (upto +1) if it lies between the camera and the 3D point, and a negative value (upto -1) if it lies behind the 3D point touching the line-of-sight. This description is then used to run our optimization algorithm on.

The primal formulation of the minimization problem can be stated as:

$$\min_u \Big\{ \int_\Omega |\nabla u| + \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|u(x) - d_i| dx \Big\}$$

The explaination and convexity analysis of this functional is given in section IV. To solve this problem, we use the first-order primal-dual algorithm introduced by Chambolle & Pock [1]. For that, we first write the primal-dual formulation of the original primal problem:

$$\min_u \max_{\|p\|_\infty \leq 1} \Big\{ - \int_\Omega u div(p) + \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|u(x) - d_i| dx \Big\}$$

where $p : \Omega \rightarrow R^3$ is the dual variable.

The algorithm consists of the gradient descent/ascent steps for u/p:

$$u^{n+1} = prox_{hist}(u^n - \tau(-divp^n))$$
$$p^{n+1} = proj_{\|p\|_\infty \leq 1}(p^n| + \sigma \nabla(2u^{n+1} - u^n))$$

where τ is the primal step and σ is the dual step size. The convergence for this algorithm is shown in seciton IV.

The functions $prox_{hist}(v)$ and $proj_{\|q\|_\infty \leq 1}(q(x))$ are defined as:

$$proj_{\|q\|_\infty \leq 1}(q(x)) = \frac{q(x)}{\max\{1, \|q(x)\|\}}$$

$$prox_{hist}(v(x)) = arg\min_u \Big\{ \frac{\|u - v(x)\|^2}{2\tau} + \lambda \sum_{i=1}^{N} h(x,i)|u - d_i| \Big\}$$

The simplification and definition of both the functions are given in [1] & [4].

III. Implementation

The dataset used for testing was the templeRing dataset from `https://vision.middlebury.edu/mview/data/`. It consists of 47 images of the model from diffferent positions on a circle alongwith the camera intrinsic and extrinsic parameters.

The simulation was written in Python 3.8 and image processing tasks were done using OpenCV library. The GPU implementation of the simulation would have took considerable time therefore it was decided to run the code on CPU only. The machine used for running the simulation is Dell G7 7588 with an 8th gen intel i7 CPU and 16GB of RAM. It has a capable Nvidia GeForce 1060 MaxQ GPU with 6GB of VRAM that will be useful in the future GPU implementation of this simulation.

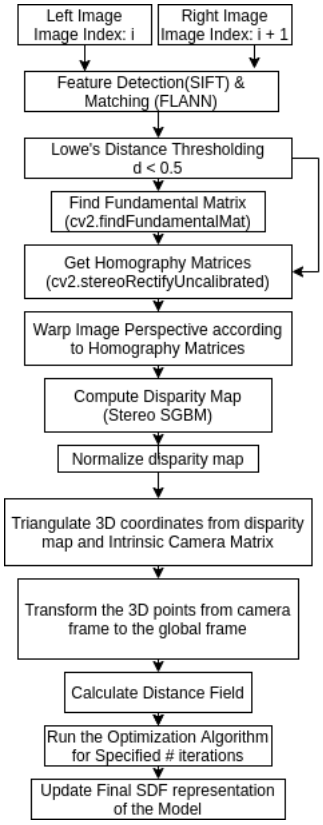


Fig. 1. Flow chart for the simulation

The flow chart for the whole procedure is given in Fig. 1.

IV. Theoretical Analysis

A. Convexity of Surface Energy Functional

To obtain the final 3D model, depth map fusion is carried out by minimizing a $TV - L^1$ energy functional modified with the histogram count of each bin i:

$$E^{TV-L^1}(u) = \int_\Omega \Big\{ |\nabla u| + \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|u(x) - d_i| \Big\} dx$$

In this functional, the first term is the measures the total variation of the function u ($\int_\Omega |\nabla u| dx = \int_\Omega \|\nabla u\|_2 dx$). The resulting function $u : \Omega \rightarrow R$ is the signed distance to the fused model. It is used to minimize the area of the level sets that define the model and thus it essentially remove noise caused by outliers in the depth map.

The other term is the L^1 term, that measures the l_1 distance of the solution to the individual distance field generated continuously from the depth maps. The term $h(x,i)$ denotes the count of how often the value d_i occured in all the distance fields generated till now at specific voxel x.

To establish the convexity of this functional, we use the zeroth order criteria for establishing convexity of any given function f:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

Let us define:

$$E(u_1) = \int_\Omega \Big\{ |\nabla u_1| + \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|u_1(x) - d_i| \Big\} dx$$

$$E(u_2) = \int_\Omega \Big\{ |\nabla u_2| + \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|u_2(x) - d_i| \Big\} dx$$

Now, applying the zeroth order criteria:

$$E(\theta u_1 + (1-\theta)u_2) = \int_\Omega \Big\{ |\nabla(\theta u_1 + (1-\theta)u_2)| + \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|(\theta u_1(x) + (1-\theta)u_2(x)) - d_i| \Big\} dx$$

Taking the first total variation term to be E_1 and the second l_1 term to be E_2:

$$E_1 = \int_\Omega |\nabla(\theta u_1 + (1-\theta)u_2)| \leq \int_\Omega \theta|\nabla u_1| + (1-\theta)|\nabla u_2|$$

$$E_2 = \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|\theta u_1(x) + (1-\theta)u_2(x) - d_i|$$

$$= \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|\theta u_1(x) + (1-\theta)u_2(x) - \theta d_i - (1-\theta)d_i|$$

$$= \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)|\theta(u_1(x) - d_i) + (1-\theta)(u_2(x) - d_i)|$$

$$\leq \lambda \sum_{i=1}^{N} \int_\Omega h(x,i)\theta|u_1(x) - d_i| + \int_\Omega h(x,i)(1-\theta)|u_2(x) - d_i|$$

Using the final results of E_1 and E_2 above, we can write:

$$E \leq E_1 + E_2$$
$$E(\theta u_1 + (1-\theta)u_2) \leq \theta E(u_1) + (1-\theta)E(u_2)$$

This establishes the convex nature of the surface energy functional.

B. Convergence of the primal-dual algorithm

The convergence of the algorithm is established in the original paper [1]. It is as follows:

Let us consider the general saddle point problem given as:

$$\min_{x \in X} \max_{y \in Y} \langle Kx, y \rangle + G(x) - F^*(y)$$

This is a primal-dula formulation of the non-linear primal problem given as:

$$\min_{x \in X} F(Kx) + G(x)$$

The dual problem is written as:

$$\max_{y \in Y} -(G^*(-K^*y) + F^*(y))$$

For our current total variational formulation:

$$F(Kx) = \int_\Omega |\nabla u| dx$$
$$G(x) = \lambda \sum_{i=1} N \int_\Omega h(x,i)|u(x) - d_i| dx$$

The algorithm that we are going to analyze is mentioned below:

Algorithm

- Initialization: Choose $\tau, \sigma > 0, \theta \in [0,1], (x^0, y^0) \in X \times Y$ and set $\bar{x}^0 = x^0$
- Iterations ($n \geq 0$): Update $x^n, y^n, \bar{x}^n$ as follows:

$$y^{n+1} = (I + \sigma \partial F^*)^{-1}(y^n + \sigma K \bar{x}^n)$$
$$x_{n+1} = (I + \tau \partial G)^{-1}(x^n - \tau K^* y^{n+1})$$
$$\bar{x}^{n+1} = x^{n+1} + \theta(x^{n+1} - x^n)$$

Let us define the primal-dual gap for the given general problem as:

$$G_{B_1 \times B_2}(x,y) = \max_{y' \in B_2} \langle y', Kx \rangle + G(x) - \min_{x' \in B_1} \langle y, Kx' \rangle - F^*(y) + G(x')$$

Now, whenever the space defined by $B_1 \times B_2$ contains a saddle point $(\hat{x}, \hat{y})$, we will have:

$$\mathcal{G}_{B_1 \times B_2} \geq 0$$

The equality occurs only when (x, y) is a saddle point.

To prove that such a saddle point exists, we use the following proof:

Rewriting the algirithm iterations in the general form:

$$y^{n+1} = (I + \sigma \partial F^*)^{-1}(y + \sigma K \bar{x})$$
$$x^{n+1} = (I + \tau \partial G)^{-1}(x^n - \tau K^* \bar{y})$$

From these equations, we get:

$$\partial F^*(y^{n+1}) \ni \frac{y^n - y^{n+1}}{\sigma} + K\bar{x}$$
$$\partial G(x^{n+1}) \ni \frac{x^n - x^{n+1}}{\tau} - K^*\bar{y}$$

such that for any $(x, y) \in X \times Y$, the $F^*(y)$ and $G(x)$ can be bounded below:

$$F^*(y) \geq F^*(y^{n+1}) + \left\langle \frac{y^n - y^{n+1}}{\sigma}, y - y^{n+1} \right\rangle + \langle K\bar{x}, y - y^{n+1} \rangle$$
$$G(x) \geq G(x^{n+1}) + \left\langle \frac{x^n - x^{n+1}}{\tau}, x - x^{n+1} \right\rangle - \langle K(x - x^{n+1}), \bar{y} \rangle$$

Summing both these inequalities and rearranging, we get:

$$\frac{\|y - y^n\|^2}{2\sigma} + \frac{\|x - x^n\|^2}{2\tau} \geq [\langle Kx^{n+1}, y\rangle - F^*(y) + G(x^{n+1})] - [\langle Kx, y^{n+1}\rangle - F^*(y^{n+1}) + G(x)] + \frac{\|y - y^{n+1}\|^2}{2\sigma} + \frac{\|x - x^{n+1}\|^2}{2\tau} + \frac{\|y^n - y^{n+1}\|^2}{2\sigma} + \frac{\|x^n - x^{n+1}\|^2}{2\tau} + \langle K(x^{n+1} - \bar{x}), y^{n+1} - y\rangle - \langle K(x^{n+1} - x), y^{n+1} - \bar{y}\rangle$$

In the last expression, all the terms except the last line are positive. Therefore, the last line is crucial in proving the convergence of the algorithm. Choosing $\theta = 0$, $\bar{x} = x^n$ and $\bar{y} = y^{n+1}$ (Arrow-Hurwicz method) in the last equation, we get the following relation for any $\beta \in (0, 1]$:

$$\begin{aligned}\langle K(x^{n+1} - \bar{x}), y^{n+1} - y\rangle - \langle K(x^{n+1} - x), y^{n+1} - \bar{y}\rangle &= \langle K(x^{n+1} - x^n), y^{n+1} - y\rangle \\ &\geq -\beta \frac{\|x^{n+1} - x^n\|^2}{2\tau} - \tau L^2 \frac{\|y^{n+1} - y^n\|^2}{2\beta} \\ &\geq -\beta \frac{\|x^{n+1} - x^n\|^2}{2\tau} - \tau \frac{L^2 D^2}{2\beta}\end{aligned}$$

where D = diam(domF^*) and $L = \|K\|$.

Continuing this for N iterations, we get the following result:

$$\begin{aligned}\sum_{n=1}^{N} [\langle Kx^n, y\rangle - F^*(y) + G(x^n)] &- [\langle Kx, y^n\rangle - F^*(y^n) + G(x)] \\ &+ \frac{\|y - y^N\|^2}{2\sigma} + \frac{\|x - x^N\|^2}{2\tau} \\ &+ \sum_{n=1}^{N} \frac{\|y^n - y^{n-1}\|^2}{2\sigma} + (1 - \beta) \sum_{n=1}^{N} \frac{\|x^n - x^{n-1}\|^2}{2\tau} \\ &\leq \frac{\|y - y^0\|^2}{2\sigma} + \frac{\|x - x^0\|^2}{2\tau} + N\tau \frac{L^2 D^2}{2\beta}\end{aligned}$$

Observe that the summation term and the norm terms in the above equation are both non-negative. Letting $x_N = (\sum_{n=1}^{N} x^n)/N$ and $y_N = (\sum_{n=1}^{N} y^n)/N$, from the convexity of G and F^*, we can write:

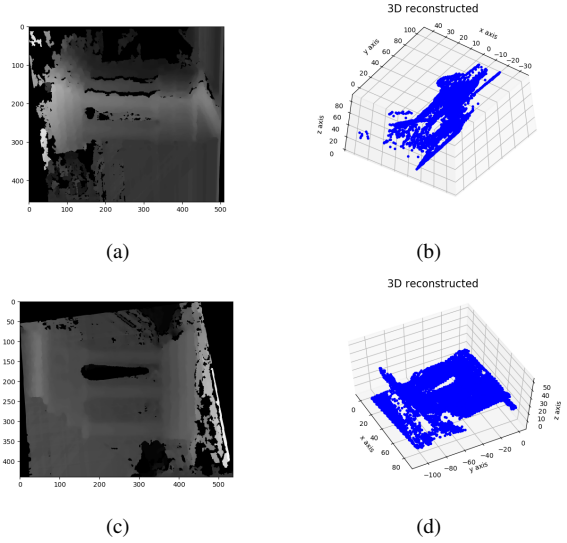
$$[\langle Kx_N, y\rangle - F^*(y) + G(x_N)] - [\langle Kx, y_N\rangle - F^*(y_N) + G(x)] \leq \frac{1}{N}\left(\frac{\|y - y^0\|^2}{2\sigma} + \frac{\|x - x^0\|^2}{2\tau}\right) + \tau \frac{L^2 D^2}{2\beta}$$

This shows that a convergence rate of $O(1/N)$ can be guaranteed within a certain error range. Also, if we choose $\tau = 1/\sqrt{N}$, we get a global $O(1/\sqrt{N})$ convergence of the gap.

More details for other cases are given in [1].

V. Simulation Results

Due to technical difficulties, I could not implement the 3D point to distance field function, and therefore could not check the final reconstruction result. The accompanying python files generate the 3D world point of the model. For the optimization algorithm, the code in the github repository for [5] was used. It contains the implementations of both the original primal-dual algorithm as well as the linesearch algorithm. Some of the stereo disparity results and coresponding 3D point clouds obtained are shown below:



(a) (b)

(c) (d)

Notice how the features of the temple are visible in the point cloud (pillars, roof and stairs).

VI. Results/Comments

For the energy functional, the choice of a TV-regularizer (total variational regularizer) poses a drawback for reconstruction since the regularization is independent of the surface normal. To improve on this, more effective regularisers can be implemented using anisotropic shape or by leveraging semantics. There are many benefits of combining semantic understanding with geometry knowledge. This is discussed in more detal in [2].

For the solver algorithm, a more efficient linesearch method can be used [5]. It requries update for only the dual (or primal) variable. It also avoids additional matrix-vector multiplications. Since it employs lesser computations and has a faster

convergence, it is more suitable for use in a real time 3D reconstruction system.

VII. Conclusion

In this paper, a convex optimization based method for 3D reconstruction of an object using multi-view geometry was presented. Some theoretical analysis for the energy functional used was discussed and its convexity was established. The proof of convergence of the used primal-dual algorithm was stated. Since the current implementation is not as fast enough, some improvements were suggested to mitigate this issue. Future works for this topic include faster generation of depth map using recent Machine Learning based techniques.

References

[1] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. J.Math. Imaging Vis., 40:120–145, May 2011.v

[2] Richard, Audrey. From Point Clouds to High-Fidelity Models-Advanced Methods for Image-Based 3D Reconstruction. Diss. ETH Zurich, 2021.

[3] C. Zach. Fast and high quality fusion of depth maps. Proc. 3DPVT, 2008.

[4] Graber, Gottfried, Thomas Pock, and Horst Bischof. "Online 3d reconstruction using convex optimization." 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, 2011.

[5] Malitsky, Yura, and Thomas Pock. "A first-order primal-dual algorithm with linesearch." SIAM Journal on Optimization 28.1 (2018): 411-432.

[6] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03, pages 1403–, Washington, DC, USA, 2003. IEEE Computer Society

[7] C. Zach. Fast and high quality fusion of depth maps. Proc. 3DPVT, 2008.

[8] https://docs.opencv.org/