Real-Time Simultaneous 3D Reconstruction and Optical Flow Estimation

Menandro Roxas Takeshi Oishi
Institute of Industrial Science, The University of Tokyo
roxas, oishi @cvl.iis.u-tokyo.ac.jp

Abstract

We present an alternative method for solving the motion stereo problem for two views in a variational framework. Instead of directly solving for the depth, we simultaneously estimate the optical flow and the 3D structure by minimizing a joint energy function consisting of an optical flow constraint and a 3D constraint. Compared to stereo methods, we impose the epipolar geometry as a soft constraint which gives the search space more flexibility instead of naïvely following the epipolar lines, resulting in a correspondence that is more robust to small errors in pose estimation. This approach also allows us to use fast dense matching methods for handling large displacement as well as shape-based smoothness constraint on the 3D surface. We show in the results that, in terms of accuracy, our method outperforms the state-of-the-art method in two-frame variational depth estimation and comparable results to existing optical flow estimation methods. With our implementation, we are able to achieve real-time performance using modern GPUs.

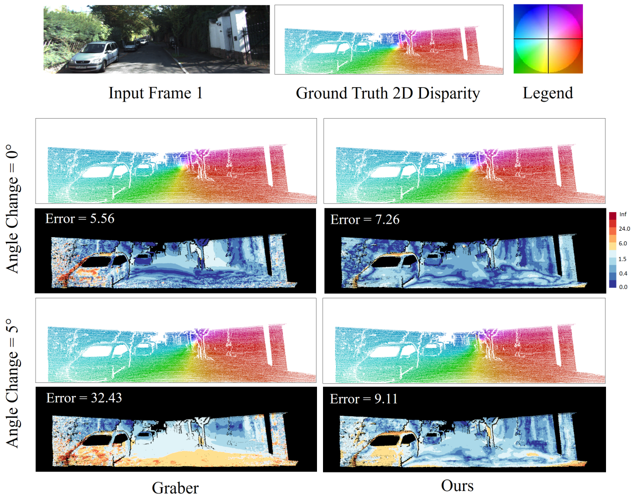

Figure 1. Robustness of applying epipolar geometry as a soft-constraint vs. classical variational stereo methods (Graber [14]). Varying the translation vector of the camera pose by 5° results in significantly higher disparity error (percentage of erroneous pixels > 3) in [14] compared to our method.

1. Introduction

In a single moving camera and static scene, the apparent motion of the pixels is characterized by the (epipolar) optical flow. Compared to general optical flow, where there is no assumption on the underlying scene structure and camera motion, the flow field is constrained along the epipolar lines which are defined by the relative camera poses. As a result, the optical flow can be used as a dense correspondence needed for 3D reconstruction. Needless to say, this relationship is embedded in motion stereo estimation methods, where the depth (or 3D structure) is solved by minimizing the brightness constancy error while imposing the epipolar geometry as a hard constraint.

In general, correspondences in stereo matching is solved by performing a search along the epipolar lines. In a variational framework, the depth can be defined through the focal length and the baseline between two cameras, and is inversely proportional to the apparent pixel motion. Assuming known intrinsic and extrinsic camera parameters, this correspondence problem reduces to a 1-D search.

In a monocular setting, the baseline is solved using egomotion estimation methods such as sparse or dense SLAM [9][10][17]. Then, the relative camera pose allows for the explicit definition of the direction vectors of the search space (i.e. epipolar lines, see [22]). In short, the resulting depth (and disparity map) is highly dependent on the correct pose estimation. However, we argue that even though the actual depth is inaccurate due to wrong pose estimation, the disparity can still be accurately estimated (see Figure 1)

To address this issue, we propose to slightly decouple the correspondence problem and the depth estimation by imposing the epipolar geometry as a soft constraint. This enables us optimize the direction vectors of the search space by allowing the disparity (or optical flow) to have isotropy while still depending on epipolar geometry (in contrast to naïvely following the epipolar lines).

We do this by minimizing a joint energy function consisting of an optical flow constraint and a 3D constraint. The 3D constraint relates the flow vectors and the 3D points through the relative camera matrices while implicitly imposing the epipolar geometry. This technique also results in the explicit definition of the 3D points, instead of depth, which allows us to use additional shape-based regularization directly on the 3D structure, while still being able to utilize existing improvements in optical flow estimation.

In this paper, we present a variational method in minimizing the joint energy function embedded in an iterative framework. Our approach results in a more accurate 3D reconstruction compared to the classical variational stereo method. We also present an implementation of our method that achieves real-time performance.

1.1. Related Work

Depth estimation using stereo vision (binocular) had been a subject of a lot of research. Using stereo methods, the correspondence problem is simplified by a 1-D search which makes them viable for real time applications. However, dense matching becomes difficult when the relative position between the two cameras are not known which is the case in single moving cameras (monocular).

Variational models address this by implicitly constraining the correspondence problem using camera egomotion. The search is simplified by removing the need for constructing expensive data structures for discrete search. Instead, the search space is restricted along the epipolar lines by defining the baseline through the constrained image derivatives. Several methods have successfully used variational models in solving the depth estimation problem [22][14]. Stuhmer et al. [22] uses the TV-L1 minimization framework and extends the method to handling multiple frames. Graber et al. [14] solves the smoothness problem of the TV-L1 method by replacing the 2D total variation function with a surface smoothness constraint.

Variational stereo problem is highly related to the optical flow problem, where the flow vectors define the 2D disparity in stereo matching. Compared to stereo methods, optical flow estimation is more expensive in that the search is not limited along the epipolar lines. On one hand, optical flow is more flexible but the a trade off between the cost and robustness has to be considered.

A lot of work has been dedicated in solving the variational optical flow problem [6][26][24][23]. Compared to its discrete counterparts (data-driven [18][2][7] or patch-based [3][11][15]), some variational models have been known to work in real time applications. For example, in [25] and [27], the optical flow is estimated by also using the TV-L1 minimization problem which is highly desirable because of its real-time implementation as well as accurate results.

Large displacement is another issue in variational models. Most real-time methods use a coarse-to-fine strategy (pyramid) to address this. However, using image pyramids is not always reliable when it comes to solving large displacements, especially for highly cluttered scenes and objects with small surface area. In [6], sparse correspondence are used as an additional constraint that is easily incorporated within the variational framework. The authors used a spatially varying mask that limits the effect of the sparse correspondence only to pixels with existing matches which results in successful estimation of large motion.

Accurate dense matching methods have also been proposed [19][26]. However, these methods are time-consuming are not desirable for real-time applications. Recently though, deep learning methods that addresses both large displacement and dense matching have been proposed [8][16]. Fortunately, these methods achieves real-time performance. However, as with all deep learning methods, the network still needs to be trained on ground truth data based on a very specific application or environment to be accurate.

Even though optical flow is assumed to be isotropic, which is efficient only within dynamic scenes, epipolar constraints similar to stereo methods have been proposed for the variational model. Valgaerts et al. [24] shows that the optical flow direction can be constrained along the epipolar lines by also estimating the fundamental matrix defining the relative pose between two cameras. Similar to our approach, this method applies the epipolar geometry as a soft constraint but our method also explicitly outputs the 3D points.

Structure-from-Motion (SfM) is a more direct estimation of the 3D structure compared to stereo matching. Most methods uses sparse correspondence between several cameras and simultaneously solves the camera poses and the 3D points. However, SfM methods are useful only for offline applications because of the multiple frame requirement. Nevertheless, Becker et al. [4] proposed to jointly solve the camera pose and depth for two frames for high speed cameras on moving vehicles. The authors use optical flow as motion observation in jointly estimating the depth and camera egomotion. Compared to our method, their approach does not refine the optical flow based on the epipolar geometry. A related method, focused on variational camera calibration was proposed by Aubry et al. [1]. In this case, the joint photometric and geometric energy is minimized resulting in camera extrinsics and dense correspondence, which are then used for reconstruction. In contrast, our method combines dense correspondence estimation and reconstruction, assuming that the camera poses are known. Moreover, our method allows for 3D surface regularization to be applied directly during energy minimization, resulting in local 3D surface smoothness that is still conformant with the dense correspondence (unlike when the regularization is applied after dense matching, which breaks the 2D regularization.)

In this work, we focus on solving the reconstruction problem for two frames. Although there are a lot of multi-view methods in existence which gives more accurate reconstruction results, we leave the extension and comparison to the multi-view problem for future work.

1.2. Overview

We first present the variational framework in Section 2 where we introduce the joint estimation of the optical flow and 3D structure. Then, we elaborate the real-time implementation in Section 3 and present the experiments, results and comparison in Section 4. Finally, we conclude this paper in Section 5.

2. Simultaneous 3D Reconstruction and Optical Flow Estimation

Our method relies on simultaneously minimizing the optical flow constraints (brightness constancy error, 2D regularization, large displacement handling) and the 3D constraints (reprojection error and surface regularization). We define our energy term for two views with known camera poses, with the only assumption that the camera translation is non-zero (not purely rotational motion).

Given two images of a static scene, $I, I' : \Omega \rightarrow \mathbb{R}^+$, taken from a moving camera with known intrinsic matrix K, we define the forward optical flow from I to I' of a pixel $\mathbf{x}$ in the image domain $\Omega \in \mathbb{R}^2$ as $\mathbf{u}$; the 3D point of each $\mathbf{x}$ as $\mathbf{X} \in S$, where $S \subset \mathbb{R}^3$ is the reconstructed surface; and, the camera matrix of I' (with respect to I) as $P = K[R|\mathbf{t}]$ where $[R|\mathbf{t}]$ is the relative camera pose.

Our objective is to find $\mathbf{u} = (u, v)$ and $\mathbf{X} = (X, Y, Z)$ for every $\mathbf{x} = (x, y)$ that minimizes the energy function:

$$\underset{\mathbf{u},\mathbf{X}}{\arg\min} \;\; F(\mathbf{x}, \mathbf{u}, \mathbf{X}) + G(\mathbf{x}, \mathbf{u}) \tag{1}$$

where F is the 3D constraint, consisting of a data term and a surface smoothness term; and G is the optical flow constraint. For simplicity, we will drop the $\mathbf{x}$ in the notations since all terms are spatially dependent on $\mathbf{x}$. We will detail the above function in the following sections.

2.1. Optical Flow Constraint

For the optical flow energy, we extended the TV-L1 optical flow functional described in [25] and added the large displacement constraint presented in [6]. We use this technique because of the existence of its minimizer that can be implemented in real-time. The modified function is as follows.

Given I and I', we define the optical flow energy function as:

$$\begin{aligned} G(\mathbf{u}) = \lambda \psi_I \left(I' \left(\mathbf{x} + \mathbf{u} \right) - I \left(\mathbf{x} \right) \right) \\ + \psi_{tv} \left(\mathbf{u}_{tv} \right) + \frac{\alpha_{tv}}{2} \|\mathbf{u} - \mathbf{u}_{tv}\|^2 \\ + \frac{\alpha_{sm}}{2} \|\mathbf{u} - \mathbf{u}_{sm}\|^2 \end{aligned} \tag{2}$$

where ψ_I is the $L1$ penalty function and ψ_{tv} is the isotropic total variation function. $\mathbf{u}_{sm}$ is the sparse optical flow value which can be solved using sparse matching methods. λ, α_{tv}, α_{sm} are weighting parameters that control the contribution strength of each function in the energy minimization. Specifically, α_{tv} is the relaxation parameter of the total variation, which when set to a high value allows for the minimum energy when $\mathbf{u}$ and $\mathbf{u}_{tv}$ are almost equal. On the other hand, α_{sm} is a sparse sampling mask that is set to zero for pixels without $\mathbf{u}_{sm}$ values, and to a positive real number otherwise.

2.2. 3D Constraint

The 3D constraint consists of a data term and a 3D smoothness term. The data term implicitly imposes the epipolar geometry and relates the 3D points and the optical flow through the camera matrices, while the smoothness term serves as the regularizer. The 3D constraint is expressed as:

$$F(\mathbf{u}, \mathbf{X}) = F_{data}(\mathbf{u}, \mathbf{X}) + F_{ms}(\mathbf{X}) \tag{3}$$

Given a set of correspondences between I and I' and the camera matrix P, the underlying 3D structure can be solved by minimizing the reprojection error. Using the optical flow $\mathbf{u}$, we can map a dense correspondence $\mathbf{x} \rightarrow \mathbf{x}'$ by assigning $\mathbf{x}' = \mathbf{x} + \mathbf{u}$ and redefine the error using $\mathbf{u}$. With this in mind, we define the data term as the sum of the reprojection errors and is expressed as:

$$F_{data}(\mathbf{u}, \mathbf{X}) = \|d\left(\mathbf{x}, P_0\mathbf{X}\right)\|^2 + \|d\left(\mathbf{x} + \mathbf{u}, P\mathbf{X}\right)\|^2 \tag{4}$$

where $d()$ is the reprojection error between the 3D point $\mathbf{X}$ and $\mathbf{x}$ through the cameras P and P_0. $P_0 = K[\mathbf{I}|\mathbf{0}]$ is the camera matrix of image I also defined as the origin $\mathbf{0}$ with identity matrix $\mathbf{I}$ as rotation.

For the 3D smoothness constraint, we use the proposed minimal surface regularizer in [14]. Given the surface S, parameterized by the image domain Ω, the tangential vectors $\mathbf{X}_x, \mathbf{X}_y$ of the infinitesimal surface dS at point $\mathbf{X}$ can be solved by the partial differentiation of $\mathbf{X}$ with respect to x and y. The infinitesimal area dA on the reconstructed surface S at point $\mathbf{X}$ is then defined on the parametric domain Ω as:

$$dA = \sqrt{det\mathbf{I_p}}d\mathbf{x} \tag{5}$$

where $\mathbf{I_p}$ is the metric tensor defined as:

$$\mathbf{I_p} = \begin{pmatrix} \langle X_x, X_x \rangle & \langle X_x, X_y \rangle \\ \langle X_x, X_y \rangle & \langle X_y, X_y \rangle \end{pmatrix} \tag{6}$$

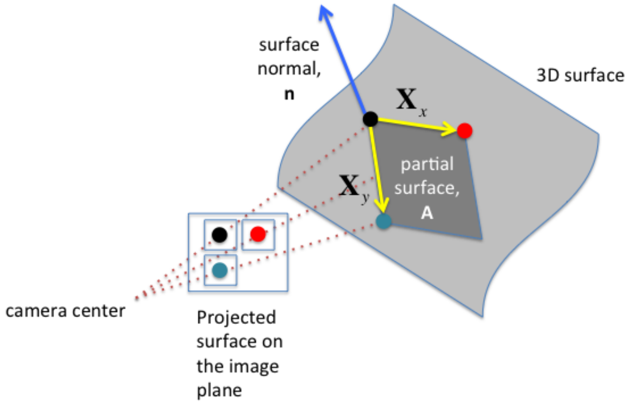

Figure 2. Minimal surface regularizer.

Our goal is to minimize the total area of the reconstructed surface, which is achieved by integrating (5) across the whole image. As in [14], the solution is equivalent to minimizing the total length of the surface normals $\|\mathbf{n}\| = \|\mathbf{X_x} \times \mathbf{X_y}\|$. From here onwards, we deviate from [14] by directly defining this cross product using the solved partial derivatives of $\mathbf{X}$. The minimal surface regularizer then becomes:

$$F_{ms}(\mathbf{X}) = \lambda_{ms}\sqrt{(Y_xZ_y-Z_xY_y)^2+(Z_xX_y-X_xZ_y)^2+(X_xY_y-Y_xX_y)^2} \tag{7}$$

2.3. Optimization

Since the reprojection error is a function of $\mathbf{u}$ and $\mathbf{X}$, it is easier to minimize (1) if we decouple this $\mathbf{u}$ from the optical flow constraint. To do this, we introduce a handler $\mathbf{u}_{pj}$ for F and impose the constraint $\mathbf{u} = \mathbf{u}_{pj}$. Modifying our main function, we get:

$$\underset{\mathbf{u},\mathbf{u}_{pj},\mathbf{X}}{\arg\min}\ F(\mathbf{u}_{pj},\mathbf{X}) + G(\mathbf{u}) + \frac{\alpha_{pj}}{2}\|\mathbf{u} - \mathbf{u}_{pj} + \mathbf{s}^k\|^2 \tag{8}$$

where $\mathbf{s}^k$ is an iteration variable [13]. We then use Alternating Direction Method to minimize the above function.

Solve for* $\mathbf{u}$*:

We first hold $\mathbf{u}_{pj}$, $\mathbf{X}$ and $\mathbf{s^k}$ constant and minimize the optical flow constraint:

$$\underset{\mathbf{u}}{\arg\min}\ G(\mathbf{u}) + \frac{\alpha_{pj}}{2}\|\mathbf{u} - \mathbf{u}_{pj} + \mathbf{s}^k\|^2 \tag{9}$$

The solution to (9) is a combination of thresholding and primal-dual decomposition. Substituting (2) to (9), we first solve for $\mathbf{u}_{tv}$ by minimizing:

$$\underset{\mathbf{u}_{tv}}{\arg\min}\ \psi_{tv}\left(\mathbf{u}_{tv}\right) + \frac{\alpha_{tv}}{2}\|\mathbf{u} - \mathbf{u}_{tv}\|^2 \tag{10}$$

which is the ROF [20] denoising problem and solved by following [25].

To solve for $\mathbf{u}$, we minimize:

$$\underset{\mathbf{u}}{\arg\min}\ \lambda\psi_I\left(I'\left(\mathbf{x}+\mathbf{u}\right) - I\left(\mathbf{x}\right)\right) + \frac{\alpha_{tv}}{2}\|\mathbf{u} - \mathbf{u}_{tv}\|^2 + \frac{\alpha_{sm}}{2}\|\mathbf{u} - \mathbf{u}_{sm}\|^2 \tag{11}$$

The problem in (11) can be solved using the thresholding scheme:

$$\mathbf{u} = \frac{\alpha_{pj}(\mathbf{u_{pj}} - \mathbf{s^k}) + \alpha_{tv}\mathbf{u}_{tv} + \alpha_{sm}\mathbf{u}_{sm}}{\alpha_{pj} + \alpha_{tv} + \alpha_{sm}} + TH(\mathbf{u}_{tv}) \tag{12}$$

The thresholding operation, $TH(\mathbf{u}_{tv})$, is defined as:

$$TH(\mathbf{u}_{tv}) = \begin{cases} -\rho(\mathbf{u}_{tv})\frac{\nabla I'}{|\nabla I'|^2}, & \text{if } |\rho(\mathbf{u}_{tv})| \leq \beta \\ \frac{\nabla I'}{\alpha_{tv}+\alpha_{pj}+\alpha_{sm}}, & \text{if} \rho(\mathbf{u}_{tv}) < \beta \\ -\frac{\nabla I'}{\alpha_{tv}+\alpha_{pj}+\alpha_{sm}}, & \text{if} \rho(\mathbf{u}_{tv}) > \beta \end{cases} \tag{13}$$

where $\rho(\mathbf{u}_{tv})$ is the linearized residual of the brightness constancy error (i.e. $I_xu_{tv} + I_yv_{tv} + I_t$). I_x and I_y are the image derivatives of I in the x and y directions, respectively, while $I_t = I' - I$. The threshold limit is defined as $\beta = \frac{\lambda|\nabla I'|^2}{\alpha_{tv}+\alpha_{pj}+\alpha_{sm}}$.

Solve for* $\mathbf{u}_{pj}$*:

Holding $\mathbf{u}$, $\mathbf{X}$, and $\mathbf{s^k}$ constant, we then solve for $\mathbf{u}_{pj}$. Since the smoothness term is independent of $\mathbf{u}_{pj}$, we get:

$$\underset{\mathbf{u}_{pj}}{\arg\min}\ \lambda_f F_{data}(\mathbf{u}_{pj}) + \frac{\alpha_{pj}}{2}\|\mathbf{u} - \mathbf{u}_{pj} + \mathbf{s}^k\|^2 \tag{14}$$

To solve (14), we first assign a temporary variable $\mathbf{u}_c$ as the flow vector between $\mathbf{x}$ and the reprojected 3D points $P\mathbf{X}$ on the image domain: $\mathbf{u}_c = P\mathbf{X} - \mathbf{x}$. We call $\mathbf{u}_c$ as the reprojected optical flow. $F(\mathbf{u}_{pj})$ can then be expressed as:

$$F_{data}(\mathbf{u}_{pj}) = \|\mathbf{u}_{pj} - \mathbf{u}_c\|^2 \tag{15}$$

The solution to (14) then becomes the weighted mean between $\mathbf{u} + \mathbf{s}^k$ and $\mathbf{u}_c$:

$$\mathbf{u}_{pj} = \frac{\lambda_f\mathbf{u_c} + \alpha_{pj}(\mathbf{u} + \mathbf{s^k})}{\lambda_f + \alpha_{pj}} \tag{16}$$

Solve for* $\mathbf{X}$*:

Given $\mathbf{u}$ and $\mathbf{u}_{pj}$, we then solve for $\mathbf{X}$:

$$\underset{\mathbf{X}}{\arg\min}\ F_{data}(\mathbf{X}) + F_{ms}(\mathbf{X}) \tag{17}$$

We rewrite the reprojection error as a linear function of $\mathbf{X}$ which defines four least squares terms:

$$\begin{aligned} F_{data}(\mathbf{X}) =& \|[x\mathbf{p_0^3} - \mathbf{p_0^1}]^T \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \|^2 + \\ & \|[y\mathbf{p_0^3} - \mathbf{p_0^2}]^T \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \|^2 + \\ & \|[(x+u)\mathbf{p^3} - \mathbf{p^1}]^T \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \|^2 + \\ & \|[(y+v)\mathbf{p^3} - \mathbf{p^2}]^T \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} \|^2 \end{aligned} \tag{18}$$

where $P_0 = [\mathbf{p_0^1}\,\mathbf{p_0^2}\,\mathbf{p_0^3}]^T$ and $P = [\mathbf{p^1}\,\mathbf{p^2}\,\mathbf{p^3}]^T$. By doing so, (17) can be solved trivially using Euler-Lagrange.

Update* $\mathbf{s}^k$*:

The last step of the alternating direction method is to update the iteration variable $\mathbf{s}^{k+1}$:

$$\mathbf{s}^{k+1} = \mathbf{s}^k + \mathbf{u} - \mathbf{u}_{pj} \tag{19}$$

3. Implementation

The optimization method in Section 2.3 is embedded in a coarse-to-fine iterative framework. We initialize all optimization variables to zero, except for hard constraints such as the camera pose P and the sparse matching $\mathbf{u}_{sm}$. To solve these values, we opt for publicly available real-time implementations that can be combined with our method. Nevertheless, there are plenty of methods that can perform a more accurate pose estimation or a cheaper sparse matching. The techniques that we used here can be easily replaced as our method is not restrictive.

3.1. Large Displacement Handling

For sparse matching, we use FlowNet2-CSS [16] implementation that is publicly available. We use the *CSS version because we don't need small displacement handling as it is already considered in our method. Furthermore, the *CSS version is much faster and allows us to perform higher iterations of our method while still achieving real-time results. To further decrease the processing time of the FlowNet2-CSS, we first scale the input image down before feeding to the network. Then, the output is scaled back up to the actual size and sampled at constant intervals using a sparse mask α_{sm}. This allows the result to further fit our proposed joint constraints.

3.2. Pose Estimation

To estimate the camera matrix P, we use the initial optical flow result of the FlowNet2-CSS as a dense correspondence. We use this initial estimate as input to the fundamental matrix estimation using Least Median Squares (LMedS) method [28], which also handles outlier rejection. With the given intrinsic camera parameters K, we solve the essential matrix and decompose it to get the relative pose. We then set the first camera position as the world center. It is not necessary to solve the actual scale of the 3D structure because the solved 3D points will be scaled back to the image domain after reprojection (see next section).

3.3. Coarse-to-Fine Approach

We implement the coarse-to-fine technique by building image pyramids with scaling factor $\alpha > 0.5$. In this strategy, the reconstruction part of the method needs to be adjusted for every level, l, of the pyramid. For a given camera matrix P, scaling only affects its intrinsic parameter K. Using $\eta = \frac{1}{\alpha}$, we can express K_{l+1} as:

$$K_{l+1} = \begin{bmatrix} \eta f_{xl} & 0 & \eta c_{xl} \\ 0 & \eta f_{yl} & \eta c_{yl} \\ 0 & 0 & 1 \end{bmatrix} \tag{20}$$

where (f_{xl}, f_{yl}) is the camera focal lengths and (c_{xl}, c_{yl}) is the image center of K_l.

We handle the scaling of the 3D points through the reprojected optical flow $\mathbf{u}_c$. Instead of directly increasing the resolution of the 3D surface, we first reproject $\mathbf{X}$ to the image domain and then solve for $\mathbf{u}_c$. Then, we scale $\mathbf{u}_c$ in the same manner as $\mathbf{u}$.

For each level of the pyramid, we embed the solution in Section 2.3 in an iterative framework until a tunable number of iterations (see Algorithm 1).

As in [5], we use warping technique to improve the estimation efficiency. For every pyramid level, we perform one warping of the input images using the initial $\mathbf{u}$ from the coarser level and solve the iteration problem on the differential flow vector $\mathbf{du}$. After each level, we add $\mathbf{du}$ back to $\mathbf{u}$ and scale the vectors accordingly.

We implemented our method on two GTX 1080 GPUs. One GPU handles the FlowNet2-CSS network and the other performs the pose estimation and our iterative method. We set the iteration at 100 which we used for the results shown in the next section. For an image input size of 1024x512 our method outputs 3D points and optical flow frames at $41ms$ (using 100 iterations, including the pose estimation). Adding the processing time of the FlowNet2-CSS, which is at $51ms$ for the scaled image input, the entire estimation is done at $10.8fps$.

4. Results and Comparison

In this section, we will first show the robustness of our method to small errors in pose estimation compared to variational stereo method. Then, we will detail the performance and results of our method and compare with existing state-of-the art methods for both optical flow [26][16]

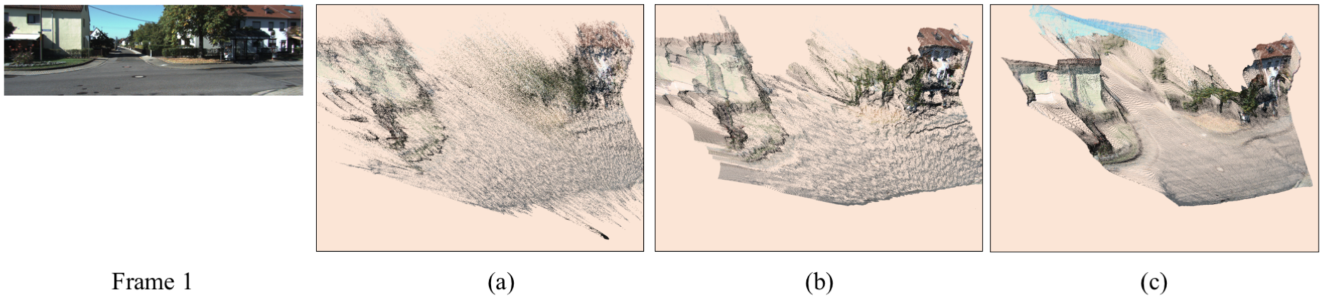

Figure 3. Reconstruction results (point cloud). (a) Reconstruction from the correspondences obtained from the optical flow estimation without the 3D regularizer [1]. (b) Applying 3D regularization after reconstruction. (c) Reconstruction using our method.

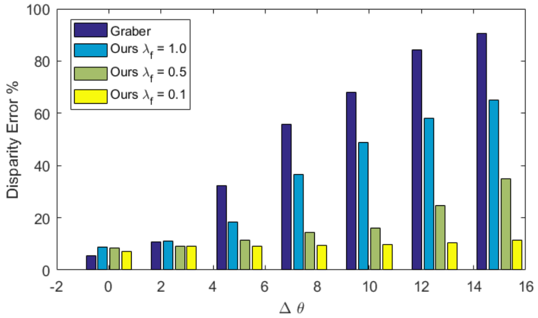

Figure 4. Disparity error vs. pose change.

and variational depth estimation [14]. For this experiment, we used the monocular pairs of images from KITTI2012 [12], which contains ground truth optical flow and depth map. We also used the stereo pairs from ETH3D [21] which contains ground truth pose and disparity map. ETH3D also contains image pairs in challenging setup such as illumination changes.

Algorithm 1 Algorithm for two-frame simultaneous 3D reconstruction and optical flow estimation.

```
Require: $I, I'$
  solve $P$
  solve $image\,pyramid$
  while $l < max\,level$ do
    $k = 0$
    while $iter > niters$ do
      solve for $\mathbf{u}_{tv}$ (10)
      solve for $\mathbf{u}$ (11)
      solve for $\mathbf{u}_{pj}$ (16)
      solve for $\mathbf{X}$ (17)
      update $\mathbf{s^k}$ (19)
      $k = k + 1$
    end while
    solve $\mathbf{u}_c$
    upsample $\mathbf{u}, \mathbf{u}_c$
    $l = l + 1$
  end while
```

Method	AEE
DeepFlow	4.48
FlowNet2-CSS	3.55
Ours	4.21

Table 1. AAE (Average endpoint error) on the KITTI2012 dataset comparison of the optical flow results among DeepFlow, FlowNet2, and our method. Our method slightly degrades the result of the FlowNet2-CSS due to errors in pose estimation.

		Graber		Ours	
		$\tau > 1$	$\tau > 3$	$\tau > 1$	$\tau > 3$
KITTI2012	000068		92.72		11.42
	000081		54.59		16.13
	000090		33.81		17.70
	000109		19.18		13.51
	000134		29.90		12.43
ETH3D	delivery-area-1l	19.38	2.379	0.840	0.012
	delivery-area-2l	38.310	2.132	2.210	0.0
	electro-1l	72.813	48.894	6.859	0.0
	facade-1s	16.805	0.738	1.310	0.191
	forest-1s	32.308	25.325	10.151	4.956
	playground-1l	69.499	55.122	15.622	1.154
	terrace-1s	80.545	53.065	2.333	0.0
	terrains-1s	96.039	90.844	3.954	0.0

Table 2. Comparison of depth results from [14] and our method on selected KITTI2012 and ETH3D dataset showing Out-Noc metric τ.

4.1. Robustness to Pose Error

To test the robustness of our method to errors in pose estimation, we vary the translation vector of the estimated camera pose via a rotation around the y-axis ($\Delta\theta$) from 0° to 15°. We plot the resulting disparity error (percentage of erroneous pixels > 3 units) for our method and compared them with [14] in Figure 4. From here, we can say that our method is able to achieve less error in disparity. More-

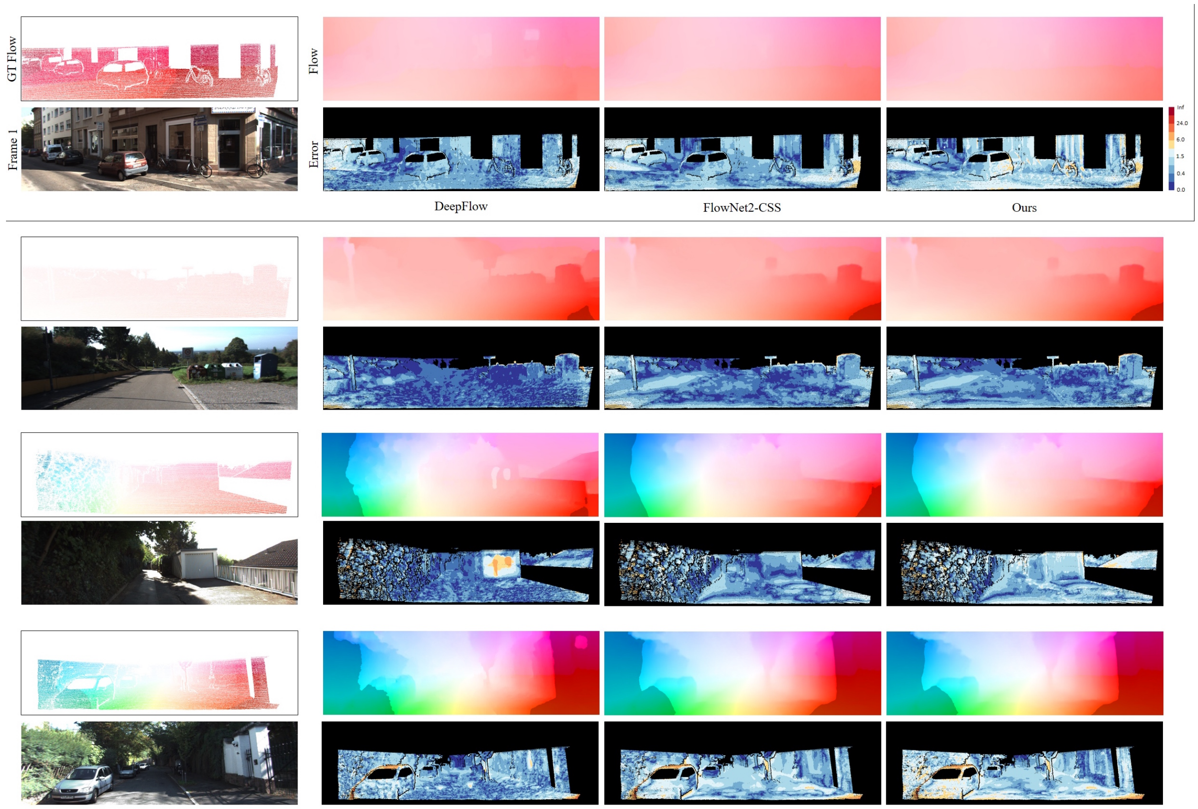

Figure 5. Optical flow and Out-Noc (percentage of erroneous pixels in non-occluded areas) results on the monocular training set of the KITTI 2012 for DeepFlow[26], FlowNet2-CSS[16] and our method.

over, the dependency of the disparity matching can be tuned through the weighting parameter, λ_f. A lower value means that the disparity is ignoring the epipolar geometry and only depending on the optical flow constraints. A sample visualization of the error is shown in Figure 1.

4.2. Optical Flow

To evaluate the optical flow, we compare our results with DeepFlow [26], which is a variational method that uses dense matching prior from DeepMatching [19], and FlowNet2 [16] which is a recent deep learning method. For both methods, we used the publicly available implementation provided by the authors. For FlowNet2, we used the *CSS version, which performed the best among many of its variants in the KITTI2012 benchmark.

We show the estimation results in Figure 5 for sample pairs with the error metric measuring the percentage of erroneous pixels (> 3) in non-occluded areas (Out-Noc). We also present the average endpoint errors (AEE) of the three methods in Table 1. From the results, our method performs better than DeepFlow in both accuracy and efficiency. Our method is also comparable to the FlowNet2-CSS results, with slightly lower accuracy. This added error is a result of errors in pose estimation which mostly affects pixels closest to the epipoles.

4.3. Reconstruction

To evaluate the reconstruction, we simply converted the 3D points to depth and compared with [14]. For this comparison, we use the same error metric (Out-Noc) as with the optical flow, considering the erroneous pixels $\tau > 1$ and $\tau > 3$ units. We use ETH3D and KITTI2012 to compare the depths and present a subset in Table 2. From the results, we can see that our method significantly outperforms [14] with estimated pose (KITTI2012) and with given ground truth pose (ETH3D). The primary reason for the significant gap in performance is due to the large displacement constraint embedded in our method. We show the actual reconstruction results in Figure 3. We compare our method with a modified version of [1] for two frames (Figure 3a). Our method achieves better smoothness because the 3D regularization is embedded in the iterative framework.

5. Conclusion and Future Work

We introduced an iterative optimization method for simultaneously estimating the optical flow and reconstruct-

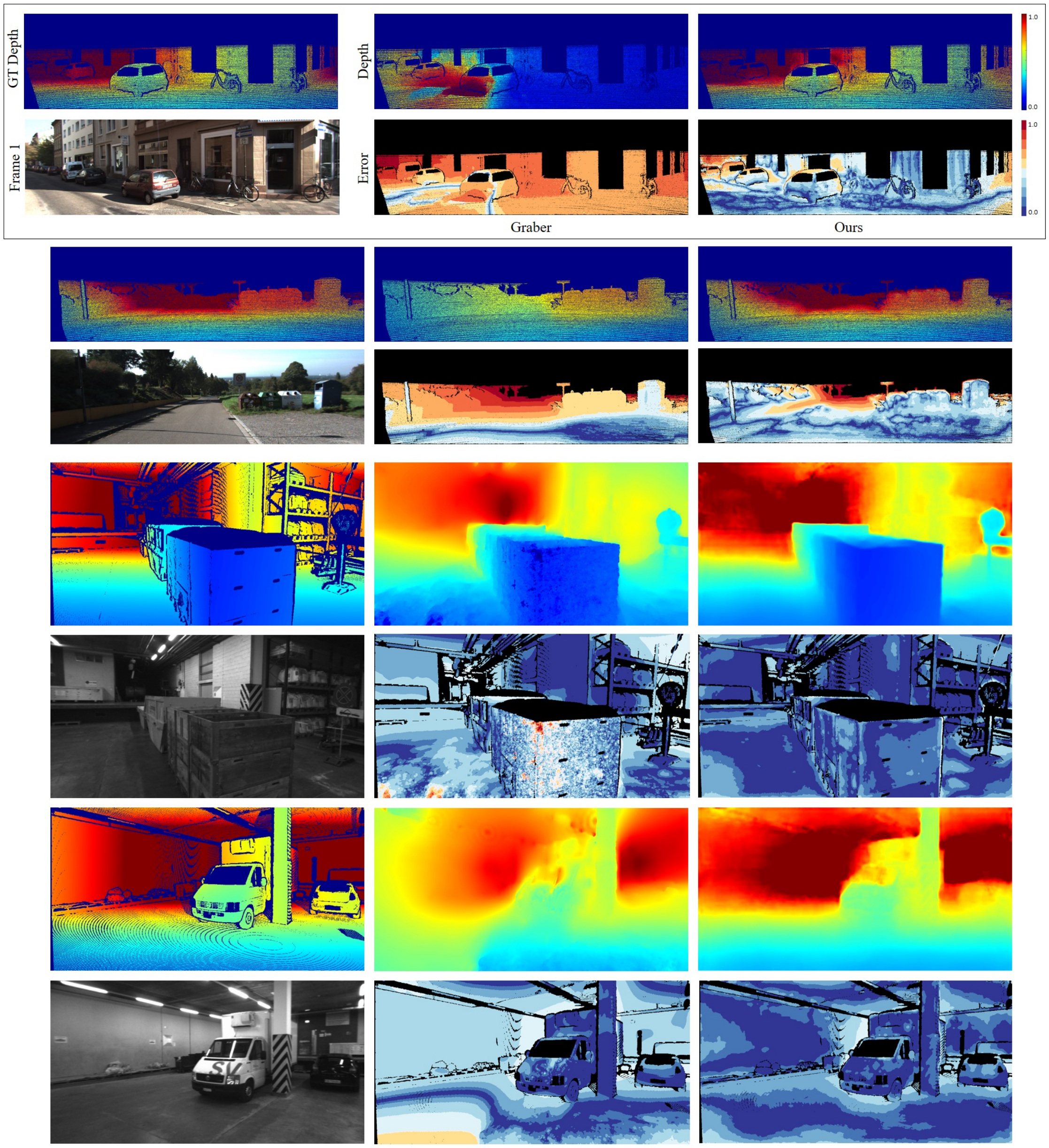

Figure 6. Comparison of depth (normalized color) between Graber [14] and our method using similar estimated (KITTI2012) and ground truth (ETH3D) pose. From top to bottom: KITTI2012 000068, 000081; ETH3D delivery-area-1l, delivery-area-2l.

ing the 3D surface. From the results, we showed that our method outperforms state-of-the-art variational depth estimation method in terms of accuracy. We also achieved comparable results with existing variational and learning-based optical flow estimation methods for outdoor static environments.

For future work, our method can be easily extended to multi-views. Since we explicitly defined the reprojection error in our energy function, the camera extrinsics and 3D structure can be separately optimized as was done in existing multi-view SfM methods, with the added advantage of also simultaneously refining the dense correspondence. Another possible direction is to simultaneously solve the optical flow, 3D geometry and camera extrinsics, which would combine our method, [1], and the classical bundle adjustment.

Furthermore, since our method works in real-time, we plan to extend this to applications such as outdoor augmented reality for moving vehicles.

References

[1] M. Aubry, K. Kolev, B. Golduecke, and D. Cremers. Decoupling photometry and geometry in dense variational camera calibration. In *ICCV*, 2011.

[2] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015.

[3] L. Bao, Q. Yang, and H. Jin. Fast edge-preserving patchmatch for large displacement optical flow. In *CVPR*, 2014.

[4] F. Becker, F. Lenzen, J. Kappes, and C. Schnorr. Variational recursive joint estimation of dense scene structure and camera motion from monocular high speed traffic sequences. *IJCV*, 105:269–297, 2013.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[6] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *TPAMI*, 33(3):500–513, 2011.

[7] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu. Large displacement optical flow with nearest neighbor fields. In *CVPR*, pages 2443–2450, 2013.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, V. GOlkov, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[9] J. Engel, T. Schops, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV*, 2014.

[10] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *ICCV*, 2013.

[11] D. Gadot and L. Wolf. Patchbatch: a batch augmented loss for optical flow. In *CVPR*, 2016.

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] T. Goldstein, X. Bresson, and S. Osher. Geometric applications of the split bregman method: Segmentation and surface reconstruction. *Journal of Scientific Computing*, 45(1):272–293, October 2010.

[14] G. Graber, J. Balzer, S. Soatto, and T. Pock. Efficient minimal-surface regularization or perspective depth maps in variational stereo. In *CVPR*, 2015.

[15] Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *CVPR*, 2016.

[16] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[17] R. Newcombe, S. Lovegrove, and A. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011.

[18] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[19] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deepmatching: Hierarchical deformable dense matching. *IJCV*, 20(3):300–323, 2016.

[20] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D.*, 60:259–268, 1992.

[21] T. Schops, J. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pellefeys, and A. Geiger. A multi-view stereo benchmark with high resolution images and multi-camera videos. In *CVPR*, 2017.

[22] J. Stuhmer, S. Grumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *DAGM*, 2010.

[23] D. Sun, S. Roth, and M. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

[24] L. Valgaerts, A. Bruhn, and J. Weickert. A variational model for the joint recovery of the fundamental matrix and the optical flow, 2008.

[25] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for tv-l1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, 2008.

[26] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, pages 1385–1392, 2013.

[27] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *DAGM*, 2007.

[28] Z. Zhang. Determining the epipolar geometry and its uncertainty: a review. *IJCV*, 27(2):161–198, 1998.