Dense depth estimation from image pairs

Lucas Payne
Supervisor: Richard Green
Department of Computer Science
University of Canterbury
Christchurch, New Zealand
lcp35@uclive.ac.nz

***Abstract*—We compare variations of a variational method for dense depth reconstruction from pairs of images. The investigated class of methods are based on the minimization of a highly non-linear functional, which, given a depth map estimate for camera one, measures the reprojection error between the corresponding pixels in cameras one and two, assuming brightness constancy. Important in this method is the use of a total-variation regularizer term, minimized with a method of Chambolle. We assume that we have an accurate (and constant) intrinsic camera matrix, our images are undistorted, and, importantly, that we have accurate camera pose estimations. The main contribution of this paper is a small framework for the visualisation of iterative methods for this problem. We have used this framework for the comparison of a simple brute-force method with a more sophisticated regularized scheme.**

—todo: numerical result

***Index Terms*—depth estimation, depth map, ROF, stereo, total variation, variational methods**

I. Introduction

In this paper, we compare variations of the method of Cremers [2] for dense depth reconstruction from grayscale images. In particular we focus on the case of image pairs. We assume (as in [2]) that we have an accurate (and constant) intrinsic camera matrix, our images are undistorted, and, importantly, that we have accurate camera pose estimations. The methodology is based on the minimization of a highly non-linear functional, which, given a depth map estimate for camera one, measures the reprojection error between the corresponding pixels in cameras one and two, assuming brightness constancy, as is assumed for the Horn-Schunck algorithm [8] for dense optical flow. Important in this method is the use of a total-variation regularizer term, which penalizes noise in the depth map while retaining depth discontinuities. To minimize this functional, we use the method of Chambolle [1] — designed for Rudin-Osher-Fatemi [7] denoising — combined with the splitting scheme of Zach et al. [9], which in their paper is applied to the problem of dense optical flow.

Following a short presentation of some important variational algorithms, we describe the method of Cremers [2] and the total-variation optimization of Chambolle [1]. Lastly, we present our main contribution: a small framework for the visualisation of iterative methods for the problem of dense depth-map estimation from image pairs. We have used this framework for the comparison of a simple brute-force (and ineffective) method with a more sophisticated regularized scheme. —todo: numerical results, discussion of future work

II. Dense variational methods

Here we give a short presentation of early variational methods used in computer vision — the Horn-Schunck and Rudin-Osher-Fatemi algorithms — and a more recent related method of Zach et al. [9] The methods outlined here serve as the algorithmic lineage of the method of Cremers et. al. [2]. See the TUM lecture series by Cremers [3], freely available online, for detailed descriptions of the mathematical underpinnings and fundamental algorithms.

A. The Horn-Schunck method for dense optical flow estimation with a regularized variational minimization.

A well-known 1981 paper on optical flow estimation by Horn and Schunck [8] introduced what is now called the "Horn-Schunck method", one of the first widespread variational algorithms used by the computer vision community (See [12] for an introduction to the problem of optical flow estimation.) The Horn-Schunck method, when introduced, was unique in that it computes a *dense* field of motion vectors. The method is usefully formulated as a continuous optimization problem. Given a stream of grayscale images $I^t : [0,1]^2 \to \mathbb{R}$, we would like to compute, for each point in the image at time t, a (space and time) velocity vector $(u, v, 1)$ which which minimizes the directional derivative

$$\nabla I^t \cdot (u, v, 1) = \frac{\partial I^t}{\partial x}u + \frac{\partial I^t}{\partial y}v + \frac{\partial I^t}{\partial t}.$$

This effectively finds the most likely "motion vector" which connects this point, as time goes on, to a point in a subsequent image with the closest intensity value. We can then formulate the functional

$$E(u,v) = \int_0^1 \int_0^1 \left(\frac{\partial I^t}{\partial x}u(x,y) + \frac{\partial I^t}{\partial y}v(x,y) + \frac{\partial I^t}{\partial t}(x,y)\right)^2 dx\, dy, \tag{1}$$

and our algorithm is simply the optimization problem

$$\min_{u,v} E(u,v). \tag{2}$$

At this point, as in all variational optimization methods, the essential conceptual work is done, and the rest of the work is in the choice of optimization strategy, and our method of discretization. However, there is a fatal flaw in (2). Suppose that the brightness constancy assumption is fully satisfied; that is, we have $\nabla I^t \cdot (u, v, 1) = 0$ for ground truth motion vector

$(u, v, 1)$. We immediately see that this is one linear equation in two variables,

$$u \frac{\partial I^t}{\partial x} + v \frac{\partial I^t}{\partial y} = - \frac{\partial I^t}{\partial t},$$

and that for any u^*, v^* such that

$$u^* \frac{\partial I^t}{\partial x} + v^* \frac{\partial I^t}{\partial y} = 0,$$

we have another solution, $(u + u^*, v + v^*)$ (see figure **??**). This is called the *aperture problem* [13], and is due to the fact that the velocity vector field is estimated locally. The key modification of Horn and Schunck is the addition of a regularizer term which penalizes high variation of the motion field, based on the observation that, except on object boundaries and occlusions, motion vector fields should be smooth. Their modified algorithm solves the optimization problem

$$\min_{u,v} \left\{ E(u, v) + R(u, v) \right\}, \tag{3}$$

where $R(u, v)$ is the regularizer term

$$\alpha^2 \int_0^1 \int_0^1 \|u\|^2 + \|v\|^2 \, dx \, dy,$$

for regularizer parameter α (the higher α, the smoother the resulting vector field). Quadratic data and regularizer terms are chosen in order to reduce the discretized Euler-Lagrange equations to a system of linear equations solved for the global minimizer (see [14] and [3] for more information on variational calculus). A quadratic regularizer, however, disproportionately penalizes discontinuities, such as across real object boundaries. We will address this problem below.

B. The Rudin-Osher-Fatemi (ROF) model for image denoising with total variation

The problem of image denoising can be put in a global optimization framework. We care here about the form of the optimization problem, rather than the application, and the ROF model turns out to be exactly the same optimization required in the "regularization step" in [9], [2]. A cost function is formulated that penalizes "noise" and rewards closeness to the original (noisy) image. The algorithm then consists of minimizing this cost function over all possible candidate "denoised" images. In fact, this algorithm is effectively the same as the above Horn-Schunck method, utilizing a slightly different data term, with a more robust regularizer.

Let $I : [0, 1]^2 \to \mathbb{R}$ be a square grayscale image with continuous domain, and $I_{i,j}$ denote the intensity at pixel (i, j) in the sampled discrete image. One formulation of the continuous ROF cost function is

$$E(\hat{I}) = \int_0^1 \int_0^1 \frac{1}{2} \left(I(x, y) - \hat{I}(x, y) \right)^2 + \lambda \|\nabla \hat{I}(x, y)\| \, dx \, dy. \tag{4}$$

The left-hand term in the integrand is the quadratic data term, penalizing differences from the original noisy image. $\|\nabla \hat{I}(x, y)\|$ is a measure of the local variation of intensity at a point in the image.

A common finite difference approximation for $\nabla \hat{I}(x, y)$, valid for non-boundary pixels, is

$$\hat{\nabla} I_{i,j} = \left(\frac{I_{i+1,j} - I_{i-1,j}}{2\Delta x}, \frac{I_{i,j+1} - I_{i,j-1}}{2\Delta y} \right)^T, \tag{5}$$

where $\Delta x, \Delta y$ are pixel extents in the image domain. Where the original image is discontinuous, such as at edges, this finite difference vector can be very large. Furthermore, the set of pixels whose finite-difference stencils extend over discontinuities is *not* neglible in the finite approximation of integral (4). A quadratic regularizer, such as $\frac{\lambda}{2}\|\nabla \hat{I}\|^2$ will harshly penalize the appearance of sharp discontinuities, since a large value returned by a finite difference is squared. The main idea behind the ROF model is to use the non-squared "total-variation" $\lambda\|\nabla \hat{I}\|$. While this is notably more difficult to optimize (precluding the use of simple linear least squares), the final effect is a preservation of isolated discontinuities such as edges and stripe patterns, while still penalizing large patches of interior noise. Notably, the ROF model can be solved by a non-linear diffusion process, performing gradient descent to solve the Euler-Lagrange equations of the cost functional [14]. See their classic paper [7] for details, and [3] for a more recent discussion.

Fundamentally, dense variational methods such as [8] and [2] follow these same lines. First, a cost functional of a image with continuous domain is formulated, penalizing unwanted properties of the solution. This cost functional is discretized, and the algorithm outputs a discrete function (such as a denoised image, or a depth map, or an optical flow field) which minimizes the discrete cost function. The majority of the complexity is in the method used to minimize (or attempt to minimize) this cost function, which could be highly non-linear.

C. Total-variation for dense optical flow estimation

In [9], Zach et al. formulate the optical flow problem as a global optimization with non-quadratic data and regularizer terms. This gives a modification of the Horn-Schunck algorithm which is more robust to outliers and which tends to preserve motion discontinuities across object boundaries. Zach et al.'s main contribution is their method of optimization, which we will reproduce in the context of dense depth map estimation. See their paper [9] for full details in the context of dense optical flow estimation.

III. Total variation for depth map estimation from image pairs

The paper by Cremer's et al. [2] applies the method of Zach et al. [9] to the problem of dense depth-map reconstruction from collections of (grayscale) images. For simplicity, we restrict our attention to the case of two images only — see [2] for details on the generalization. We assume (as in [2]) that we have an accurate (and constant) intrinsic camera matrix, our images are undistorted, and, importantly, that we have accurate camera pose estimations. We describe here a slightly modified version of Cremer's method.

A. The optimization problem

Let C_1 and C_2 denote the two cameras, I_1 and I_2 denote the two images, and let h be the estimated depth map for C_1. Our fully non-linear cost function is

$$\begin{aligned} E(h) = \frac{\lambda}{|\Omega^*|} \int_{\Omega^*} & \|I_2(\text{reproj}(x,y)) - I_1(x,y)\| \, dx \, dy \\ & + \int_0^1 \int_0^1 \|\nabla h\| \, dx \, dy. \end{aligned} \tag{6}$$

which has TV-L^1 [9] data and regularizer terms. $\text{reproj}(x,y)$ is where the non-linear complexity is. x and y denote image coordinates in I_1, and h is used to transform these image coordinates to world-space, given the camera pose of C_1. We can then project this point to the image space of I_2. $\Omega^* \subset [0,1]^2$ is defined as

$$\Omega_0 := \left\{ (x,y) \in [0,1]^2 \mid \text{reproj}(x,y) \in [0,1]^2 \right\}. \tag{7}$$

Therefore, our data term is weighted over those pixels which have valid reprojections. Without the $1/|\Omega^*|$ factor in the data term, our optimization would likely cause h to be a constant function (giving zero regularizer cost), large enough such that all reprojections are invalid (giving zero data cost) — In our implementation, weighting by $1/|\Omega^*|$ has been effective at preventing this.

B. Splitting the cost function

Let

$$\rho(h) = \|I_2(\text{reproj}(x,y)) - I_1(x,y)\|.$$

Since the data term is non-quadratic, we cannot apply the ROF optimization method directly. However, we can introduce an auxiliary depth map h' to separate the data and regularizer terms, and introduce a penalizer for disparity between h' and h (the same method as used in [9] for optical flow). The new separable cost function is

$$\hat{E}_\theta(h,h') = \int_\Omega \lambda \|\rho(h)\| + \frac{1}{2\theta}(h-h')^2 + \|\nabla h'\|. \tag{8}$$

The parameter θ is set to some small constant. The cost function E_θ is separated into functions of h and h', and h and h' are alternately optimized in the following way:

- For h fixed, solve
$$\min_{h'} \int_\Omega \lambda \|\nabla h'\| + \frac{1}{2\theta}(h-h')^2 \, dx \tag{9}$$
using the ROF optimization methods.
- For h' fixed, solve
$$\min_{h} \int_\Omega \|\rho(h)\| + \frac{1}{2\theta}(h-h')^2 \, dx. \tag{10}$$
Since $\rho(h)$ measures point-wise reprojection error, this optimization can be done point-wise.

C. Efficiently solving the ROF step

The ROF cost function (9) can be minimized using the iterative method of Chambolle [1]. For completeness, we reproduce this method here in the context of dense depth map estimation. We provide no proofs — see [1] for full details.

D. Interpretation of the algorithm and its parameters

We are left with many parameters: regularizer parameter λ, thresholding parameter θ, chosen each iteration, and the number of iterations. The performance of our method relies heavily on the choices of these values — the preliminary results seen in figure 1c were only achieved by manual adjustment. Although, in principle, we aim to globally optimize the original cost function (**??**), our final method of optimization can be reinterpreted as a step-based algorithm, outlined here:

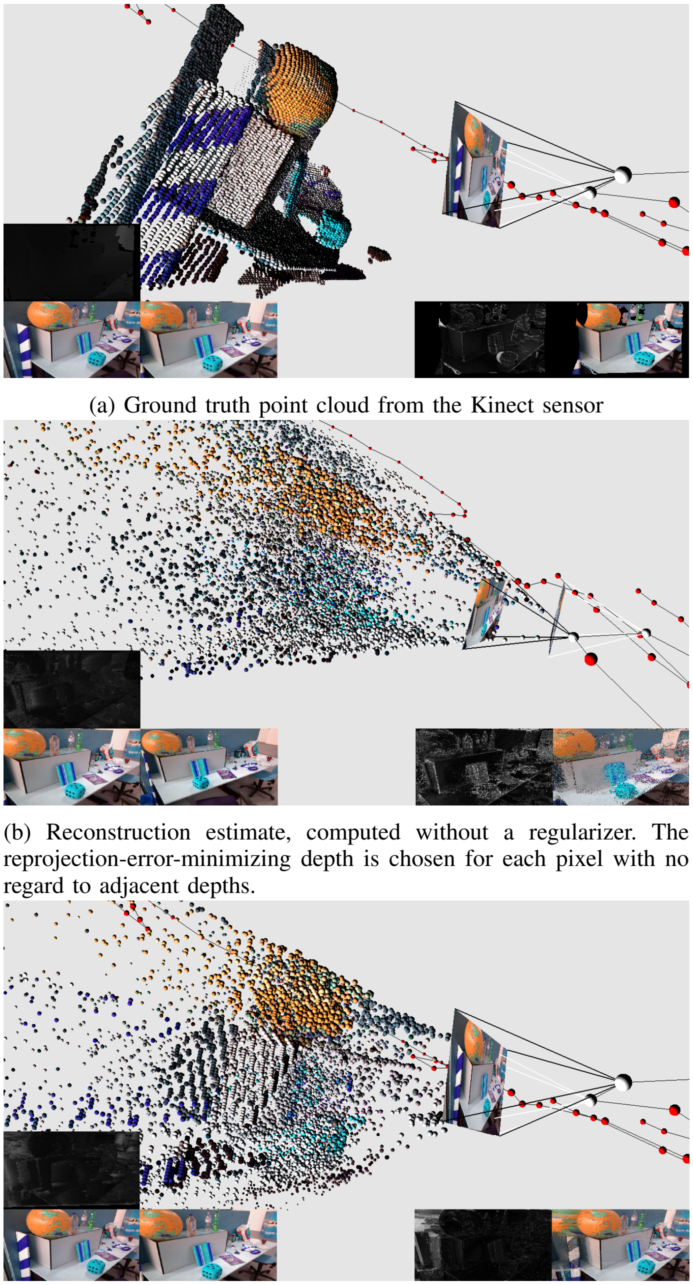
1) Choose regularizer parameter λ.
2) Choose an initial value for splitting parameter θ.
3) Initialize a depth map (say, to a constant 3 meters).
4) For each pixel in camera one, find the point on the ray which reprojects to the closest intensity in camera two. Update the depth map at this pixel to be the distance to that point.
5) Regularization (smoothing) step. Solve the ROF model to smooth the noisy depth map.
6) Repeat step 4, but additionally penalize changes to the depth map. (e.g., find the point on the ray which minimizes the sum of reprojection error and a quadratic function of the difference to the old depth value).
7) Go to step 5, unless sufficiently many iterations have been completed.

IV. Visualization and Dataset

A visual tool for rapidly prototyping new dense reconstruction methods is provided. We evaluate our method on the "freiburg3_long_office_household" RGBD dataset, which contains accurate ground-truth camera poses, and depth maps constructed using a Kinect sensor [**?**] [4]. Our tool allows these depth maps to be used as a drop-in replacement for the estimated depth maps, to get a rough comparison against the "ground truth". Figure 2b shows an example visual inspection of the output of our total-variation algorithm.

Figure 1c displays some preliminary results, using heuristically determined parameters of θ, λ, and τ, and around 25 iterations. Clearly this depth map is not satisfactory. However, it has some important qualitative features. Firstly, self-occlusion is captured for the orange globe in the reprojection image, which is not captured by the brute-force method. Secondly, the point cloud clusters generally toward the real object locations — note the orange cluster near the globe. These results are compared to the ground truth point cloud shown in figure 1a.

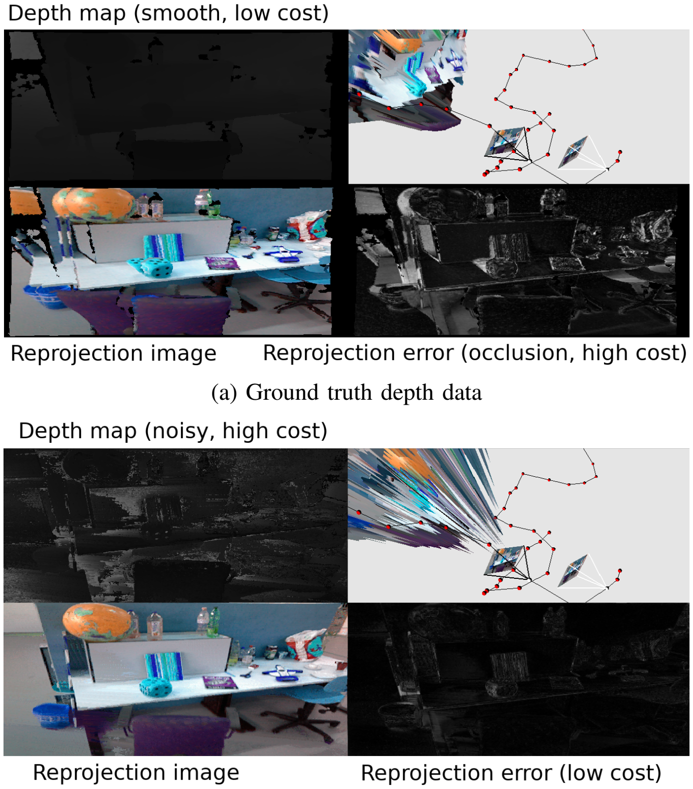
Consider the naive approach of minimizing reprojection error pointwise. For one pixel in frame 1, the depth value parameterizes a ray which is then reprojected into frame 2. A simple brute-force approach is to search frame 2, along this reprojected ray, for the color which gives minimum reprojection error. Figures 2a and 2b visualize the problem



(a) Ground truth point cloud from the Kinect sensor

(b) Reconstruction estimate, computed without a regularizer. The reprojection-error-minimizing depth is chosen for each pixel with no regard to adjacent depths.

(c) Reconstruction estimate, computed with a regularizer. Note that point clusters are preserved yet begin to form (noisy) surfaces.


Depth map (smooth, low cost)
Reprojection image
Reprojection error (occlusion, high cost)
Depth map (noisy, high cost)
Reprojection image
Reprojection error (low cost)

(a) Ground truth depth data

(b) Brute-force computed, with no regularizer

with this approach. Figure 2a displays, using the the ground truth depth map data provided by the Kinect sensor [6], anticlockwise from the bottom right:

- The reprojection image.
- The reprojection error.
- A 3D view of the depth map placed in world space.
- The depth map.

Figure 2b displays the same results when using a depth map computed with the naive brute-force algorithm outlined above.

The ground truth reprojection image contains "coloured shadows" due to occlusion. These also contribute to high (white) values in the reprojection error image. The brute-force depth map avoids this occlusion problem.

Consider a pixel in frame 1 corresponding a point which, from the perspective of camera 2, is behind the blue die. Likely, the line search in frame 2 will find an unoccluded point, of similar colour, on the table. This, however, gives an unrealistic (and noisy) depth value in general — compare the depth images in figures 2a and 2b. Clearly we would like to favour the depth map in 2a, and this is exactly the reason for the (denoising) regularizing term in (4).

Due to the freedom of the line search, the brute-force method has achieved low reprojection error (bottom right of figure 2b). With the ground truth depth map, however, the self-occlusion causes a large amount of reprojection error, which will penalize the cost function (4). We would clearly like to minimize the total reprojection error (with the data term), but also keep the depth map "smooth" (with the regularizer term), as achieved in [2]. However, it is difficult to choose a good value of λ to balance the data and regularizer.

We propose a modification to the method of [2], by removing the

V. Comparison of results

VI. Conclusion

We have shown the utility of such an algorithm-specific testing framework. It is hoped that, although our reconstructed depth maps are of much lower quality and our iterative method converges far more slowly than the state of the art reconstruction methods, it is hoped that this is merely an architectural issue.

We have visualised the meaning of the regularizer term used in [2].

A. Future work

In future research, we would like to implement the much faster, GPU-accelerated multi-grid method of Cremers [2], and perform a systematic comparison of methods from prior research on dense depth reconstruction ([**?**]). The main utility of our framework is in the visualization of the effect of parameters such as the regularizer weight λ and thresholding parameter θ. We would also like to automate comparisons of the results using the rest of the TUM RGB-D dataset [6]. A standardized measure of quality of a depth-from-image-pair algorithm will likely be of great utility. The utility of depth map estimation from colour images is largely in mobile 3D reconstruction and augmented reality applications. Planned features include an extension of our framework to support the visualisation of volumetric reconstruction algorithms ([4], [10], [11]), by fusing the depth map estimations together into a volumetric data structure.

References

References

[1] Chambolle, A. (2004). An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision, 20(1/2), 89–97.*

[2] Cremers, D., Stühmer, J., Gumhold, S., (2010). Real-Time Dense Geometry from a Handheld Camera. *Lecture Notes in Computer Science, 11–20.*

[3] Cremers, D., Variational Methods in Computer Vision, TUM Department of Infomatics, https://vision.in.tum.de/teaching/online/cvvm (online resource). (Link valid as of June 2021).

[4] Cremers, D., Steinbrücker, F., Kerl, C., Stürm, J., Large-Scale Multi-Resolution Surface Reconstruction from RGB-D Sequences. In ICCV, 2013.

[5] Cremers, D., Steinbrücker, F., Stürm, J., Volumetric 3D Mapping in Real-Time on a CPU. In ICRA, 2014.

[6] A Benchmark for the Evaluation of RGB-D SLAM Systems (J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers), In Proc. of the International Conference on Intelligent Robot Systems (IROS), 2012.

[7] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena, 60(1–4), 259–268.*

[8] Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence, 17(1–3), 185–203.*

[9] Zach, C., Pock, T., Bischof, H. (2007). A Duality Based Approach for Realtime TV–L^1 Optical Flow. *Pattern Recognition: 29th DAGM Symposium, Heidelberg, Germany. Springer. 214-223.*

[10] Curless, B., Levoy, M. (1996). A Volumetric Method for Building Complex Models from Range Images. In SIGGRAPH, 1996.

[11] R. Newcombe et al. (2011). KinectFusion: Real-Time Dense Surface Mapping and Tracking. Microsoft Research.

[12] Szeliski, R. (2010). Computer Vision: Algorithms and Applications (Texts in Computer Science). Springer.

[13] aperture problem

[14] Gelfand, I. M., & Fomin, S. V. (2000). Calculus of Variations (Dover Books on Mathematics). Dover Publications.