

Dense depth estimation from image pairs

Lucas Payne

Department of Computer Science and Software Engineering

University of Canterbury

Christchurch, New Zealand

lcp35@uclive.ac.nz

Abstract—We compare variations of the variational method of Cremers for dense depth reconstruction from grayscale images. In particular we focus on the case of image pairs. 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 ala. the Horn-Schunck algorithm for dense optical flow. 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—component, formatting, style, styling, insert

I. INTRODUCTION

In this paper, we compare variations of the variational method of Cremers [3] for dense depth reconstruction from grayscale images. In particular we focus on the case of image pairs. We assume (as in [3]) 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 ala. the Horn-Schunck algorithm [7] for dense optical flow. Important in this method is the use of a total-variation regularizer term, which penalizes noise in the image. We use the method of Chambolle [15] designed for Rudin-Osher-Fatemi [?] denoising, combined with the splitting scheme of Zach et al. [4], 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 [3] and the total-variation optimization of Chambolle [15]. 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 methods outlined here serve as the algorithmic lineage of the method of Cremers et. al. [3]. See the TUM lecture series by Cremers [13], freely available online, for detailed descriptions of the mathematical underpinnings and fundamental algorithms.

A. Dense optical flow with the Horn-Schunck method

A well-known 1981 paper on optical flow estimation by Horn and Schunck [7] introduced what is now called the “Horn-Schunck method”, one of the first widespread variational algorithms used by the computer vision community. To estimate a dense field of motion vectors,

B. Total-variation denoising with the Rudin-Osher-Fatemi (ROF) model

The problem of image denoising can be put in a global optimization framework. 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.

Let $I : [0, 1]^2 \rightarrow \mathbb{R}$ 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

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

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

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

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* negligible in the finite approximation of integral (1). 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. See their classic paper [1] for details, and [2] for a more recent discussion.

Fundamentally, dense variational methods such as (??) and (??) follow these same lines. First, a cost functional of an 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 [4], Zach et. al formulate the optical flow problem [14] 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 [4] 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 [3] applies the method of Zach et. al [4] 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 [3] for details on the generalization. We assume (as in [3]) that we have an accurate (and constant) intrinsic camera matrix, our images are undistorted, and, importantly, that we have accurate camera pose estimations. Our fully non-linear cost function is

$$E(h) = \lambda \int_{\Omega_0} \|I_1(\pi(\exp(\hat{\eta}_1)X(x, h)) - I_0(\pi(x)))\| + \|\nabla h\| d^2x, \quad (3)$$

which has TV- L^1 [4] data and regularizer terms. (—REDO reprojection notation) Let $\rho(h) = \|I_1(\pi(\exp(\hat{\eta}_1)X(x, h)) - I_0(\pi(x)))\|$. 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 new separable cost function is

$$\hat{E}_\theta(h, \hat{h}) = \int_{\Omega} \lambda \|\rho(h)\| + \frac{1}{2\theta} (h - \hat{h})^2 + \|\nabla \hat{h}\|. \quad (4)$$

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

$$\min_{h'} \int_{\Omega} \lambda \|\nabla h'\| + \frac{1}{2\theta} (h - h')^2 dx \quad (5)$$

using the ROF optimization methods.

- For h' fixed, solve

$$\min_h \int_{\Omega} \|\rho(h)\| + \frac{1}{2\theta} (h - h')^2 dx. \quad (6)$$

Since $\rho(h)$ measures point-wise reprojection error, this optimization can be done point-wise.

The ROF cost function (5) can be minimized using the iterative method of Chambolle [15]. For completeness, we reproduce this method here in the context of dense depth map estimation. We provide no proofs — see [15] for full details.

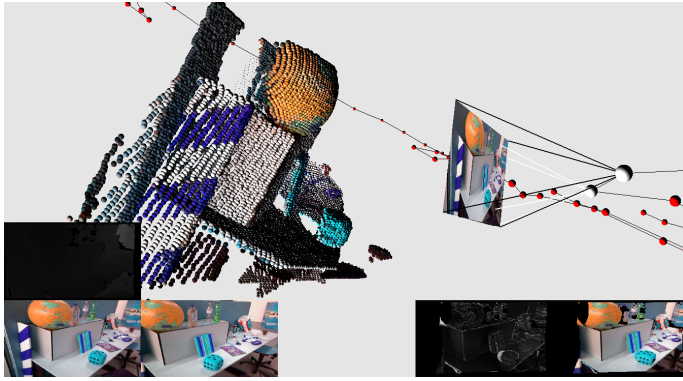
IV. VISUALIZATION AND DATASET

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 [5] [6]. 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 (ref figure) shows an example visual inspection of the output of our total-variation algorithm.

Figure [?] 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 [?].

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 [?] and [?] visualize the problem with this approach. Figure [?] displays, using the the ground truth depth map data provided by the Kinect sensor [12], anticlockwise from the bottom right:

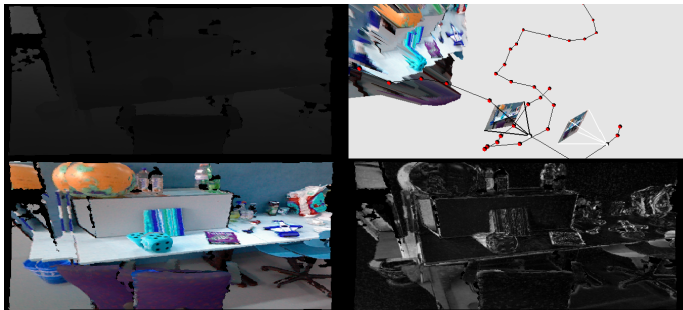
- The reprojection image.
- The reprojection error.
- A 3D view of the depth map placed in world space.
- The depth map.



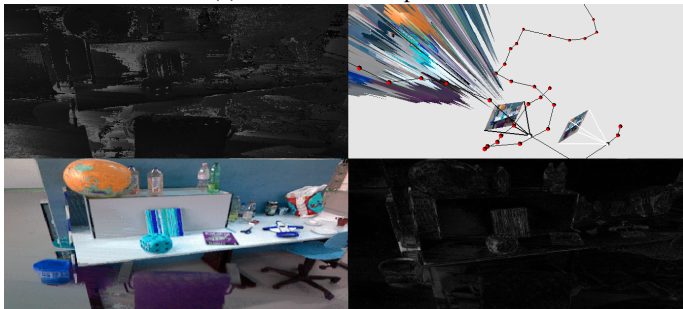
(a) Ground truth point cloud from the Kinect sensor



(b) Reconstruction estimate, computed with a regularizer



(a) Ground truth depth data



(b) Brute-force computed, with no regularizer

Figure [?] 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 to 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 [?] and [?]. Clearly we would like to favour the depth map in (2a), and this is exactly the reason for the (denoising) regularizing term in (1).

Due to the freedom of the line search, the brute-force method has achieved low reprojection error (bottom right of figure [?]). With the ground truth depth map, however, the self-occlusion causes a large amount of reprojection error, which will penalize the cost function (1). 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 [3]. 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 [3], by removing the

V. COMPARISON OF RESULTS

VI. POSSIBILITIES FOR FURTHER WORK

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 ROF
- [3] Cremers, D., Stühmer, J., Gumhold, S., (2010). Real-Time Dense Geometry from a Handheld Camera. *Lecture Notes in Computer Science*, 11–20.
- [4] Cremers, D., Variational Methods in Computer Vision, TUM Department of Informatics, <https://vision.in.tum.de/teaching/online/cvmm> (online resource). (Link valid as of June 2021).
- [5] Cremers, D., Steinbrücker, F., Kerl, C., Sturm, J., Large-Scale Multi-Resolution Surface Reconstruction from RGB-D Sequences. In ICCV, 2013.
- [6] Cremers, D., Steinbrücker, F., Sturm, J., Volumetric 3D Mapping in Real-Time on a CPU. In ICRA, 2014.
- [7] 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.
- [8] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4), 259–268.
- [9] Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1–3), 185–203.
- [10] 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.
- [11] B. Curless, M. Levoy, A Volumetric Method for Building Complex Models from Range Images. In SIGGRAPH, 1996.
- [12] R. Newcombe et al., KinectFusion: Real-Time Dense Surface Mapping and Tracking. Microsoft Research, 2011.