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*—Abstract**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

...

II. Dense variational methods

Varitional methods

A. Dense optical flow with the Horn-Schunck method

A well-known 1981 paper on optical flow estimation by Horn and Schunck **??** introduced what is now called the "Horn-Schunck method", one of the first widespread variational algorithms used by 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 $I : [0,1]^2 \to \mathbb{R}$ 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

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

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

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

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* neglible 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 **??** for details, and **??** for a more recent discussion.

Fundamentally, dense variational methods such as (**??**), (**??**) 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. See the TUM lecture series by Cremers **??**, freely available online, for more discussion.

C. Total-variation for dense optical flow estimation

?? $TV - L^1$ data and regularizer terms.

III. Total variation for depth map estimation from image pairs

The paper by Cremer's et. al **??** applies insights from the previously mentioned methods (such as the thresholding method for total-variation minimization in **??**) 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 **??** for details on the generalization. We assume (as in **??**) that we have an accurate (and constant) intrinsic camera matrix, our images are undistorted, and, importantly, that we have accurate camera pose estimations.

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

This equation takes the same form as the above optical flow cost function **??**. 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

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

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 $h_{\prime}$ are alternately optimized in the following way:

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

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 **??** **??**. 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.

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.

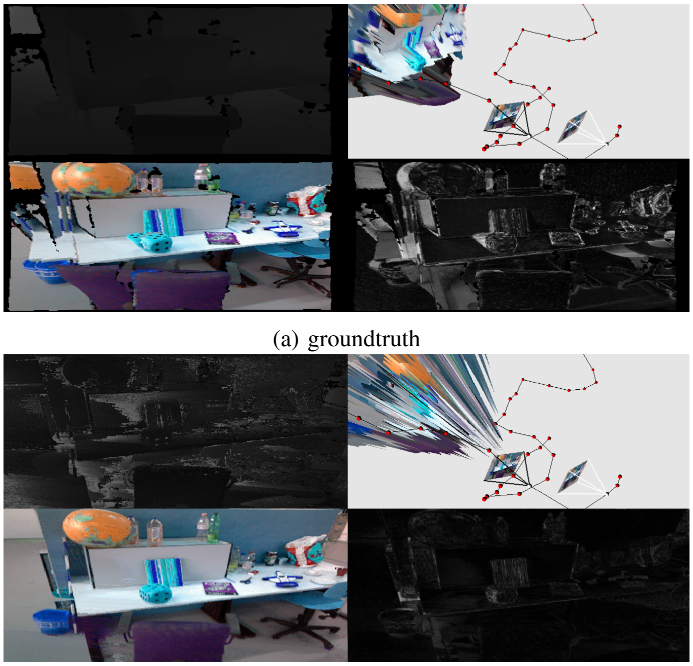
Figures 1a and 1b visualize the problem with this approach. Figure 1a displays, using the the ground truth depth map data provided by the Kinect sensor **??**, anticlockwise from the bottom right:

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

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


(a) groundtruth

(b) computed

Due to the freedom of the line search, the brute-force method has achieved low reprojection error (bottom right of figure 1b). 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 **??**. 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 **??**, by removing the

V. Comparison of results

VI. Possibilities for further work

References

...

References

[1] ROF
[2] Cremers ROF
[3] Cremers, Real-Time Dense Geometry from a Handheld Camera.
[4] Paper in Pattern recognition book
[5] Cremers, dataset
[6] Cremers, largescale use of dataset
[7] Horn, Schunck.
[8] B. Curless, M. Levoy, A Volumetric Method for Building Complex Models from Range Images. In SIGGRAPH, 1996.

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

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

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

[12] 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.

[13] variational lecture series