FAST STRUCTURE FROM MOTION FOR PLANAR IMAGE SEQUENCES

Andreas Weishaupt, Luigi Bagnato, and Pierre Vandergheynst

Signal Processing Laboratory (LTS2), Ecole Polytechnique Fédérale de Lausanne (EPFL)
Station 11, CH–1015 Lausanne, Switzerland
phone: +41 21 69 36874, email: {andreas.weishaupt,luigi.bagnato,pierre.vandergheynst}@epfl.ch

ABSTRACT
Dense three-dimensional reconstruction of a scene from images is a challenging task. Usually, it is achieved by finding correspondences in successive images and computing the distance by means of epipolar geometry. In this paper, we propose a variational framework to solve the depth from motion problem for planar image sequences. We derive camera ego-motion estimation equations and we show how to combine the depth map and ego-motion estimation in a single algorithm. We successfully test our method on synthetic image sequences for general camera translation. Our method is highly parallelizable and thus well adapted for real-time implementation on the GPU.

1. INTRODUCTION
The efficient three-dimensional recovery of a scene structure from images has been a long-term aim in computer vision. Successful methods would have a big impact on a broad range of fields such as autonomous navigation, biomedical imaging or architecture. The geometry that links the 3D structure of a scene and its projection on images has been studied thoroughly, e.g. in [5] or [4]. Traditionally, there is one main approach: from a pair of stereo images of the scene the 3D recovery is based on epipolar geometry. Such an approach has been extended to methods that handle multiple camera inputs which are now generally known as multi-view stereo methods (see [7] for a good overview and comparison). Structure from motion means the 3D scene reconstruction from images captured by a moving camera. Usually, similar methods are used in structure from motion recovery. They rely on finding pairs of corresponding points in successive images. This has the following consequences:

• The final result depends on the quality of the found correspondence. If the match is not exact the reconstruction will not be accurate.
• Finding correspondences is a computationally expensive task: dense reconstruction cannot be performed in real-time.
• For real-time reconstruction, the recovery has to be limited to some few feature points. Often, tracking of the found feature points is employed to reduce additional computation cost. Tracking introduces a second source of error for 3D reconstruction.

Another class of recent methods to obtain dense depth maps is based on the fusion of sparse depth maps by image registration techniques, e.g. [8]. Those have the advantage that traditional structure from motion systems can be employed. However, in order to provide accurate results, a large number of depth maps has to be input for such methods. Finally, in [10] it is shown how robust depth map reconstruction can be achieved from video sequence by belief propagation and bundle optimization. Even if the results are promising and the method does not need hundreds of frames to be input, it is computationally expensive and only destined for off-line processing.

Bagnato et al. [2, 1] has shown how to recover dense depth maps from omnidirectional image sequences by employing a variational framework. The approach does not have the usual drawbacks found for the methods above:

• Dense depth maps can be obtained without finding correspondences and combining sparse depth maps.
• One depth map can be recovered by using only two successive input images.
• The framework can be implemented for a dense frame-by-frame recovery in real-time.

In this paper, we will show how to modify this approach in order to handle regular images obtained by a planar image sensor. In Section 2, we derive a general projection model that relates depth and camera motion. The model is nonlinear due to the central projection on the image plane and we show how it can be linearized. In Section 3, the linearized projection can be used in a TV-L1 optimization framework for depth from motion reconstruction. In Section 4, we solve the camera ego-motion estimation problem. As a last step, in Section 5 we combine both, the ego-motion and the depth from motion estimation into a complete structure from motion framework for planar image sequences. In section 6, we separately evaluate the ego-motion and depth from motion estimation on synthetic images, and then we evaluate the complete algorithm with quite remarkable results.

2. MOTION IN PLANAR IMAGES
We model the camera movement during acquisition of two successive frames by the rigid 3D translation vector \( t = (t_x, t_y, t_z)^T \). Consequently, during camera motion, a point \( p = (X, Y, Z)^T \) becomes \( p' = p - \Delta p = p - t \) where \( \Delta p \) denotes the relative camera motion. We parametrize \( p \) by \( p = d(r)e_s \) where \( r = (x, y, f) \) is a point on the planar sensor, \( f \) the focal distance and \( d(r) \) the distance or depth of the optical center to a point in the scene. The pinhole camera model and motion is shown in Figure 1. We denote \( Z(r) = 1/d(r) \) the inverse depth or depth map. A point on the sensor plane can be obtained by central projection:

\[ r = \frac{||r||p}{d(r)} = ||r||Z(r)p. \] (1)

In Figure 2 we have a side view of the camera model and motion, as well as projections on the sensor plane and on the parallel object plane. Based on Eq. 1, we can derive a projection model that links camera movement, depth of the scene
and optical flow. The optical flow \( u \) is defined as the apparent motion of brightness pattern between two images. The central projection on the object plane parallel to the sensor plane is given by:

\[
t_p = \frac{r_c \cdot r + ||r||Z(r)t}{||r||Z(r)t + ||r||Z(r)c} - \frac{r}{||r||Z(r)}. \tag{2}
\]

Let us define Eq. 2 as the parallel projection. The optical flow can be approximated by the following projection on the sensor plane:

\[
u = r_c \cdot r + ||r||Z(r)t \tag{3}
\]

Eq. 3 shows the nonlinear dependency of the estimated optical flow on the depth map \( Z(r) \) as well as on the translation \( t_c \) perpendicular to the sensor plane. In this nonlinear form, it is difficult to include the projection in a variational framework. Nevertheless, combining Eqs. 2 and 3 we find a linearized relationship between the parallel projection and the optical flow:

\[
u = ||r||Z(r)t_p. \tag{4}
\]

### 3. TV-L\(_1\) DEPTH FROM MOTION

We assume for the moment that we know the camera translation parameters \( t \) for two successive frames \( I_0 \) and \( I_1 \). Furthermore, we assume that the brightness does not change between these images. Using the definition of optical flow and the projection in Eq. 4, we can express the image residual \( \rho(Z) \) as in [6]:

\[
\rho(Z) = I_1(x + u_0) + \nabla I_1^T (||r||Zt_p - u_0) - I_0. \tag{5}
\]

A depth map \( Z = Z(r) \) can be obtained by solving the following optimization problem [2]:

\[
Z^* = \arg\min_Z \sum_{x \in D} |\nabla Z| + \lambda \sum_{x \in D} \rho(Z, I_0, I_1), \tag{6}
\]

where \( D \) is the discrete domain of pixels and \( x \) their position on the image. The left term in Eq. 6 represents the regularization term. Here we set it to the TV norm of \( Z \) which imposes a sparseness constraint on \( Z \) and acts edge-preserving. The right term is the data term which we set to the image residual as defined in Eq. 5. We have chosen the robust \( L_1 \) norm as it has some advantages when compared to the usually employed \( L_2 \) norm [9]. Eq. 6 is not a strictly convex optimization problem and thus hard to solve. From [2] and [9] we know that a convex relaxation can be formulated:

\[
Z^* = \arg\min_{Z} \sum_{x \in D} |\nabla Z| + \frac{1}{2\theta} \sum_{x \in D} (V - Z)^2 + \lambda \sum_{x \in D} \rho(V), \tag{7}
\]

where \( V \) is a close approximation of \( Z \) and for \( \theta \to 0 \) we have \( V \to Z \). We solve Eq. 7 using an alternative two-step iteration scheme:

1. For fixed \( Z \), solve for \( V \):

\[
V^* = \arg\min_{V} \sum_{x \in D} \frac{1}{2\theta} (V - Z)^2 + \lambda \sum_{x \in D} \rho(V). \tag{8}
\]

2. For fixed \( V \), solve for \( Z \):

\[
Z^* = \arg\min_{Z} \sum_{x \in D} |\nabla Z| + \frac{1}{2\theta} \sum_{x \in D} (V - Z)^2. \tag{9}
\]

Eq. 8 can be solved by the following soft-thresholding:

\[
V = Z + \begin{cases} 
\lambda \theta ||\nabla I_1^T t_p \rho(Z)/||\nabla I_1^T||_p^2 & \text{if } \rho(Z) < -\lambda \theta ||\nabla I_1^T t_p \rho(Z)/||\nabla I_1^T||_p^2 \\lambda \theta ||\nabla I_1^T t_p \rho(Z)/||\nabla I_1^T||_p^2 & \text{if } \rho(Z) > \lambda \theta ||\nabla I_1^T t_p \rho(Z)/||\nabla I_1^T||_p^2 
\end{cases} \tag{10}
\]

In order to solve Eq. 9, the dual formulation of the TV norm can be exploited. It is given by: \( TV(Z) = \max\{p \cdot \nabla Z : ||p|| \leq 1\} \). With the introduced dual variable \( p \), Eq. 9 can be solved iteratively by the Chambolle algorithm [3, 2]:

\[
p^{n+1} = \frac{p^n + \tau V(V \cdot p^n - V/\theta)}{1 + \tau \nabla V \cdot p^n - V/\theta}. \tag{11}
\]

In the discrete domain the stability and properties of the solution depends on the implementation of the differential operators. In Eq. 11, \( V \) represents the discrete gradient operator and the scalar product with \( V \) represents the discrete divergence operator as defined in [3]. From Eq. 11, the depth map can be recovered by \( Z = V - \theta V \cdot p \). Furthermore, the depth positivity constraint has to be imposed on the recovered depth map, i.e. if \( Z(r) < 0 \) we set \( Z(r) \to 0 \). In order to provide global convergence and to handle different levels of detail in the depth map \( Z \) we propose solving Eq. 7 using a multi-scale resolution approach. This means that we
The Jacobian matrix of a function $f: \mathbb{R}^n \to \mathbb{R}^m$ is a matrix whose entries are the partial derivatives of the function. For example, if $f$ is a function of two variables $x$ and $y$, the Jacobian matrix is:

$$
J_f = \begin{pmatrix}
\frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\
\frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y}
\end{pmatrix}
$$

where $f_1$ and $f_2$ are the components of $f$. The Jacobian matrix is used in various applications, including optimization algorithms, where it is used to approximate the function locally around a point. In the context of computer vision, the Jacobian matrix is used to relate the change in the image to the change in the 3D scene.

The article discusses the use of downsampling and upsampling in the context of estimating ego-motion and depth. The authors propose an approach that alternates between estimating depth and ego-motion, which is found to be effective for certain types of sequences. The experimental results show that the proposed approach can achieve good performance in both depth and ego-motion estimation.

The authors also provide a mathematical formulation for the problem of depth and ego-motion estimation, which involves solving a system of equations that relate the changes in the image to the changes in the 3D scene. The formulation involves the use of partial derivatives, which are computed using the chain rule of calculus.

The article concludes with a comparison of the proposed approach with existing methods, and the authors report that their method performs well in terms of both accuracy and speed.

In summary, the article provides a detailed analysis of the problem of depth and ego-motion estimation, with a focus on the use of downsampling and upsampling. The authors propose an effective approach that alternates between estimating depth and ego-motion, and they report good performance in both tasks. The mathematical formulation and experimental results are presented in detail, providing a comprehensive understanding of the problem.

5. Joint Depth and Ego-Motion Estimation

For a complete depth map reconstruction from input images, we must show how to combine the depth from motion estimation described in Section 3 and the ego-motion estimation described in Section 4. Since both parts rely on each other, it is very likely that we can combine them by performing alternating depth and ego-motion estimation. We find that it is best to include the alternation scheme in the multi-scale approach:

1. At the coarsest resolution level $L$, we initialize $t^L$ by zero and $Z^L$ by some small constant. We can first solve for $Z^L$ as explained in Section 3. Since the ego-motion parameters are zero the estimated depth map will be very flat.

2. With the flat depth map as input we estimate the motion parameters according to Section 4. The estimated motion parameters will be close to the true underlying motion parameters since the estimation is well constrained with an approximately constant depth map as input.

3. Given the estimated motion parameters $k+1 t$ and the depth map $k+1 Z$, we first estimate the optical flow $u_0 = \frac{\partial r}{\partial Z} t$, then we compute the depth map at level $k+1 Z$.

4. From the refined depth map $k+1 Z$, we compute the motion parameters $k+1 t$.

5. Steps 3 and 4 are repeated until the finest resolution is reached and the final depth map $k Z$ is obtained.

6. Results

In order to verify our approach, we use synthetic images of size $512 \times 512$ and ground truth depth maps generated by ray-tracing of a 3D model of a living room. We have generated multiple sequences for various types of camera translation, i.e. for movement parallel and perpendicular to the image plane as well as for linear combinations of both. The purpose is to evaluate first ego-motion estimation and depth from motion seperately.

We run the ego-motion estimation with ground truth depth maps on the different sequences and we obtain the translation vector estimates as listed in Table 1. For simplicity we only show the mean and standard deviation of the normalized vectors.

We evaluate the depth from motion part by using the ground truth translation vectors as inputs. We normalize the input images which is convenient for comparing the used pa-
rameters. In our experiments, we use 5 levels of resolution with a constant scale factor of 2 from level to level. The functional splitting parameter $\theta$ is set to 0.05. $\theta \to 0$ means approaching the true TV-$L_1$ model and thus better accounting for discontinuities in the image which in general is very welcome for depth map reconstruction. However, the smaller $\theta$ is set, the more iterations are needed for satisfying convergence. Setting $\theta = 0.05$ proves to be a good choice that does not require too much iterations while providing an acceptable recovery of edges in most cases. We find empirically that the regularization parameter $\lambda$ should be set such that the product $\lambda \theta$ lies between 1 and 10 percent of the gray-level range of the input images. We use $\lambda = 0.5$. Increasing or decreasing $\lambda$ too much results respectively in over or under-regularization and thus in very inaccurate depth maps.

Two successive images of an input sequence for motion in distinct x and z direction are shown in Figure 3. Ground truth depth maps for the evaluation of the depth from motion part as well as the recovered depth maps are shown in Figure 4.

The complete joint ego-motion and depth from motion estimation algorithm is evaluated on the same synthetic sequences as above. In Figure 5 ground truth and recovered depth maps are shown for the complete joint algorithm. We use the same parameters and remarks as made for separate ego-motion and depth from motion estimation.

Our framework only uses simple, spatially well localized operations including element-wise multiplication, gradient and warping. It does not include so-called parallel reduction operations like finding the global minimum or maximum of an image. Consequently, it is well adapted to implementation on a parallel architecture such as the graphics processing unit (GPU). Furthermore, our algorithm only needs to process two input frames in order to compute a depth map. Thus, it is well adapted for real-time performance. Our current prototype GPGPU implementation reaches a performance of 5 frames per second.

7. CONCLUSION AND FUTURE WORK

This paper presented a variational framework for dense depth map recovery given two successive frames obtained by a moving camera with a planar image sensor. We showed results obtained for different kinds of camera movement which are very promising given the difficulty of the subject. Since our framework is highly parallelizable and only needs two
successive images as input to compute a depth map, our GPGPU implementation reaches a performance of 5 frames per second.

Our framework works well for camera translation while we find that it is not simple to include camera rotation. This is mainly due to the limits of planar imaging, i.e. that the optical flow pattern for a small rotation is hardly distinguishable from the one issued by sensor-parallel translation. However, the camera rotation can be accurately recovered by external sensors like accelerometers which are more and more present in mobile devices.

The theoretical limits of scene depth recovery with our projection model tell us that we cannot equally well recover depth for close and far objects. However, there exist different approaches to bypass this limit, e.g. depth map fusion, voting or surface models. As future work, we will include such a method in our framework for a refined depth maps recovery and for a consistent 3D reconstruction of the scene.

Acknowledgement

The authors would like to thank Emmanuel D’Angelo for his support.

References


