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]. Traditionnally, 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 computationnally 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 additionnal 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-L_1 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 $\mathbf{t} = (t_x, t_y, t_z)^T$. Consequently, during camera motion, a point $\mathbf{p} = (X, Y, Z)^T$ becomes $\mathbf{p}' = \mathbf{p} - \mathbf{\Delta p} = \mathbf{p} - \mathbf{t}$ where $\mathbf{\Delta p}$ denotes the relative camera motion. We parametrize $\mathbf{p}$ by $\mathbf{p} = d(\mathbf{r})\mathbf{e_r}$ where $\mathbf{r} = (x, y, f)$ is a point on the planar sensor, f the focal distance and $d(\mathbf{r})$ 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 $Z(\mathbf{r}) = 1/d(\mathbf{r})$ the inverse depth or *depth map*. A point on the sensor plane can be obtained by central projection:

$$\mathbf{r} = \frac{\|\mathbf{r}\|\mathbf{p}}{d(\mathbf{r})} = \|\mathbf{r}\| Z(\mathbf{r}) \mathbf{p}. \tag{1}$$

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 parellel object plane. Based on Eq. 1, we can derive a projection model that links camera movement, depth of the scene

 ISSN 2076-1465

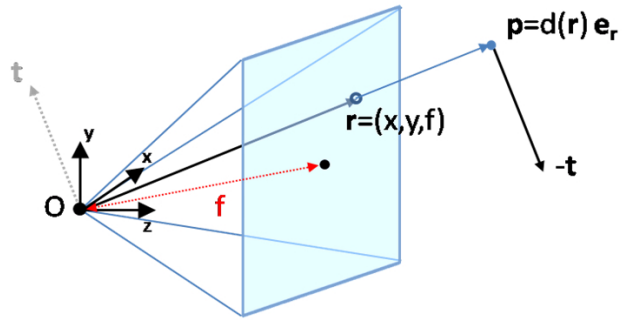

Figure 1: pinhole camera model and rigid camera motion

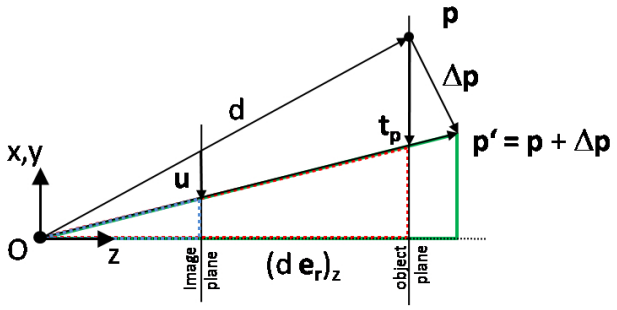

Figure 2: side view with the projections of camera motion

and optical flow. The *optical flow* $\mathbf{u}$ 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:

$$\mathbf{t_p} = \frac{r_z}{\|\mathbf{r}\|Z(\mathbf{r})} \cdot \frac{\mathbf{r} + \|\mathbf{r}\|Z(\mathbf{r})\mathbf{t}}{r_z + \|\mathbf{r}\|Z(\mathbf{r})t_z} - \frac{\mathbf{r}}{\|\mathbf{r}\|Z(\mathbf{r})}. \tag{2}$$

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

$$\mathbf{u} = r_z \cdot \frac{\mathbf{r} + \|\mathbf{r}\|Z(\mathbf{r})\mathbf{t}}{r_z + \|\mathbf{r}\|Z(\mathbf{r})t_z} - \mathbf{r}. \tag{3}$$

Eq.3 shows the nonlinear dependency of the estimated optical flow on the depth map $Z(\mathbf{r})$ as well as on the translation t_z 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:

$$\mathbf{u} = \|\mathbf{r}\|Z(\mathbf{r})\mathbf{t_p}. \tag{4}$$

3. TV-L_1 DEPTH FROM MOTION

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

$$\rho(Z) = I_1(\mathbf{x} + \mathbf{u_0}) + \nabla I_1^T(\|\mathbf{r}\|Z\mathbf{t_p} - \mathbf{u_0}) - I_0. \tag{5}$$

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

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

where D is the discrete domain of pixels and $\mathbf{x}$ 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:

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

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:

$$V^* = \arg\min_V \sum_{\mathbf{x}\in D} \frac{1}{2\theta}(V - Z)^2 + \lambda \sum_{\mathbf{x}\in D} |\rho(V)|. \tag{8}$$

2. For fixed V, solve for Z:

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

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

$$V = Z + \begin{cases} \lambda\theta\|\mathbf{r}\|\nabla I_1^T\mathbf{t_p} & \text{if } \rho(Z) < -\lambda\theta(\|\mathbf{r}\|\nabla I_1^T\mathbf{t_p})^2 \\ -\lambda\theta\|\mathbf{r}\|\nabla I_1^T\mathbf{t_p} & \text{if } \rho(Z) > \lambda\theta(\|\mathbf{r}\|\nabla I_1^T\mathbf{t_p})^2 \\ \frac{\rho(Z)}{\|\mathbf{r}\|\nabla I_1^T\mathbf{t_p}} & \text{if } |\rho(Z)| \le \lambda\theta(\|\mathbf{r}\|\nabla I_1^T\mathbf{t_p})^2 \end{cases}. \tag{10}$$

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

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

In the discrete domain the stability and properties of the solution depends on the implementation of the differential operators. In Eq. 11, ∇ represents the discrete gradient operator and the scalar product with ∇ represents the discrete divergence operator as defined in [3]. From Eq. 11, the depth map can be recovered by $Z = V - \theta\nabla \cdot \mathbf{p}$. Furthermore, the *depth positivity constraint* has to be imposed on the recovered depth map, i.e. if $Z(\mathbf{r}) < 0$ we set $Z(\mathbf{r}) \leftarrow 0$. 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 use downsampled images ${}^kI_0$ and ${}^kI_1$ of L different sizes (the

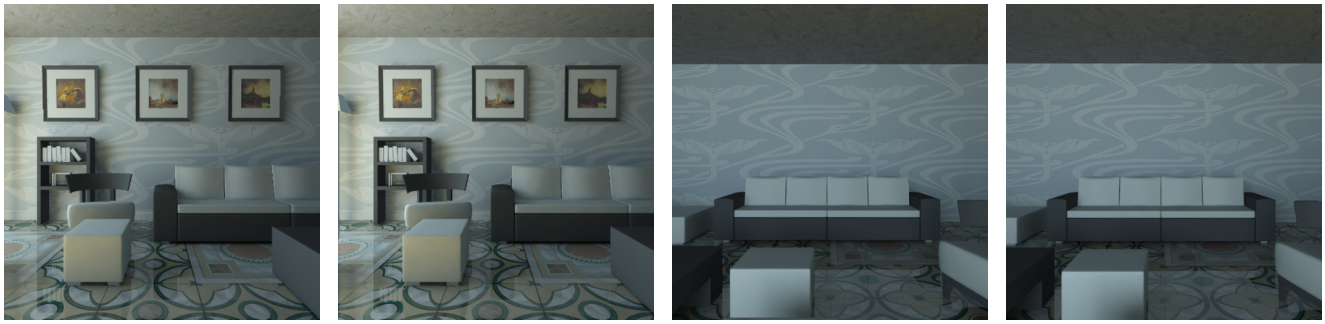

Figure 3: Some of the used input images. Left: two successive frames for movement in x direction with a translation vector $\mathbf{t} = (-0.1, 0, 0)^T$. Right: two successive frames for movement in z direction with a translation vector $\mathbf{t} = (0, 0, 0.1)^T$.

prefix k denotes the scale level). We start at the coarsest resolution, where we solve for ${}^L Z$. Then we upsample ${}^L Z$ to the next level $L-1$ and use it as input for the projection in the image residual. This can be repeated until level 0 is reached where we obtain the final depth map ${}^0 Z$.

4. EGO-MOTION ESTIMATION

Let us assume now that we have an estimate of the depth map Z. Given two successive images I_0 and I_1 we can recover the camera translation parameters by optimizing the L_2 norm of the image residual with respect to $\mathbf{t}$:

$$\sum_{\mathbf{x} \in D} \left(I_1 - I_0 + \|\mathbf{r}\| Z \mathbf{t_p}^T \nabla I_1 \right) \|\mathbf{r}\| Z \nabla I_1 = 0. \qquad (12)$$

In the special case of camera movement parallel to the sensor plane solving Eq. 12 results in the linear system $\mathbf{A}(\mathbf{x})\mathbf{b} = \mathbf{c}(\mathbf{x})$ with

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} \sum_{\mathbf{x} \in D} \|\mathbf{r}\|^2 Z^2 \left(\frac{\partial I_1}{\partial x} \right)^2 & \sum_{\mathbf{x} \in D} \|\mathbf{r}\|^2 Z^2 \frac{\partial I_1}{\partial x} \frac{\partial I_1}{\partial y} \\ \sum_{\mathbf{x} \in D} \|\mathbf{r}\|^2 Z^2 \frac{\partial I_1}{\partial x} \frac{\partial I_1}{\partial y} & \sum_{\mathbf{x} \in D} \|\mathbf{r}\|^2 Z^2 \left(\frac{\partial I_1}{\partial y} \right)^2 \end{pmatrix}$$

and

$$\mathbf{c}(\mathbf{x}) = \begin{pmatrix} -\sum_{\mathbf{x} \in D} \|\mathbf{r}\| Z \frac{\partial I_1}{\partial x} (I_1 - I_0) \\ -\sum_{\mathbf{x} \in D} \|\mathbf{r}\| Z \frac{\partial I_1}{\partial y} (I_1 - I_0) \end{pmatrix}.$$

For general camera motion, we can solve Eq. 12 by iterative methods, e.g. Levenberg-Marquardt or gradient descent: $\mathbf{x}^{n+1} = \mathbf{x}^n + \gamma \nabla E(\mathbf{x}^n)$ where $\mathbf{x}$ contains the three translation parameters and E is the energy of the image residual,

$$\frac{\partial E}{\partial x_i} = \sum_{\mathbf{x} \in D} \left(I_1 - I_0 + \nabla I_1^T \mathbf{u} \right) \nabla I_1^T \frac{\partial \mathbf{u}}{\partial x_i}.$$

The partial derivatives of $\mathbf{u}$ with respect to the motion parameters are given by the Jacobian matrix

$$\mathbf{J_u^T} = \begin{pmatrix} \frac{r_z \|\mathbf{r}\| Z}{r_z + \|\mathbf{r}\| Z t_z} & 0 \\ 0 & \frac{r_z \|\mathbf{r}\| Z}{r_z + \|\mathbf{r}\| Z t_z} \\ -r_z \|\mathbf{r}\| Z \frac{r_x + \|\mathbf{r}\| Z t_x}{(r_z + \|\mathbf{r}\| Z t_z)^2} & -r_z \|\mathbf{r}\| Z \frac{r_y + \|\mathbf{r}\| Z t_y}{(r_z + \|\mathbf{r}\| Z t_z)^2} \end{pmatrix}.$$

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 ${}^L \mathbf{t}$ by zero and ${}^L Z$ by some small constant. We can first solve for ${}^L Z$ 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.
3. Given the estimated motion parameters ${}^{k+1} \mathbf{t}$ and the depth map ${}^{k+1} Z$, we first estimate the optical flow $\mathbf{u_0} = \|\mathbf{r}\| Z(\mathbf{r}) \mathbf{t_p}$, then we compute the depth map at level ${}^k Z$.
4. From the refined depth map ${}^k Z$, we compute the motion parameters ${}^k \mathbf{t}$.
5. Steps 3 and 4 are repeated until the finest resolution is reached and the final depth map ${}^0 Z$ is obtained.

6. RESULTS

In order to verify our approach, we use synthetic images of size 512×512 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 parameters. In our experiments, we use 5 levels of resolution with a constant scale factor of 2 from level to level. The functional splitting parameter θ is set to 0.05. $\theta \to 0$ means approaching the true TV-L_1 model and thus better accounting

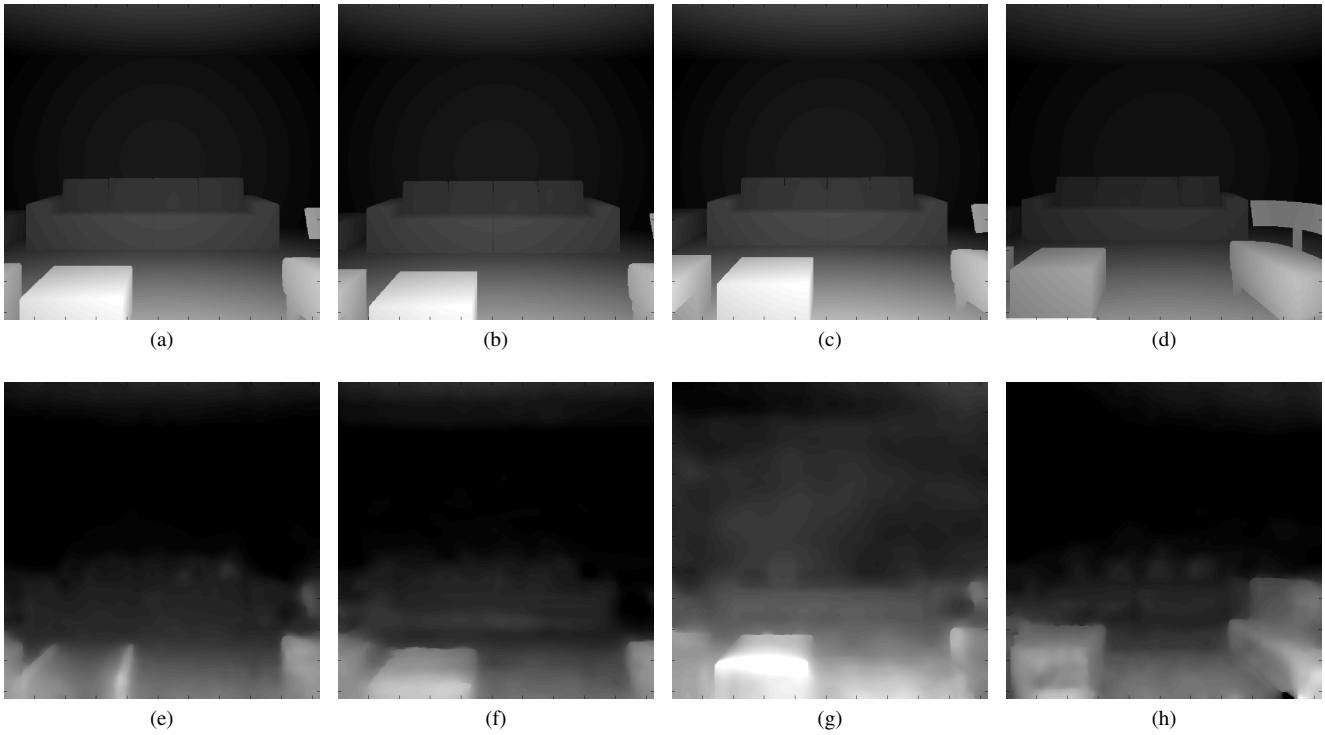

Figure 4: Depth maps. Top row: ground truth depth maps. Bottom row: recovered depth maps with the depth from motion algorithm using ground truth ego-motion. Columns from left to right: movement in x, y, z and x+y+z direction respectively. MSEs of estimated depth $d(\mathbf{r})$ are: e) 3.3, f) 2.8, g) 4.2 and h) 6.4. MSEs of the estimated depth map $Z(\mathbf{r})$ are: e) $4.5 \cdot 10^{-4}$, f) $2.7 \cdot 10^{-4}$, g) $3.1 \cdot 10^{-4}$ and h) $3 \cdot 10^{-4}$.

for discontinuities in the image which in general is very welcome for depth map reconstruction. However, the smaller θ is set, the more iterations are needed for satisfying convergence. Setting $\theta = 0.05$ proofs 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 λ 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 λ too much results respectively in over or under-regularization and thus in very inaccurate depth maps.

Two successive images of an input sequence for movement in distinct x and z direction are shown in Figure 3. Ground truth 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.

Transl.	**true t**	**mean(t)**	**std(t)**
x	(1,0,0)	(0.97,-0.02, 0.16)	(0.03, 0.03, 0.18)
y	(0,1,0)	(-0.02, 0.84, 0.43)	(0.02, 0.08, 0.35)
z	(0,0,1)	(-0.03, 0.06, 0.99)	(0.01, 0.01, 0.00)
x+z	(1,0,1)	(0.88, 0.13, 1.09)	(0.09, 0.02, 0.08)
y+z	(0,1,1)	(-0.19, 1.26, 0.38)	(0.15, 0.09, 0.47)
x+y+z	(1,1,1)	(0.39, 1.38, 0.84)	(0.13, 0.16, 0.45)

Table 1: Ego-motion estimation using nonlinear-least squares. We use a Levenberg-Marquardt solver with initial search point $\mathbf{t}^0 = (0,0,0)$.

Errors in the ego-motion estimation might be high. This is primarily due to using a zero-translation vector as starting point which results in a flat depth map at coarse resolution. The motion parameters estimation at coarse resolution is therefore almost fully constrained by the input images only. As long as the coarse input images carry enough translational information, this will result in a reasonable motion estimation. But it will result in quite large errors if this is not the case.

Our framework only uses simple, spatially well localized operations. Consequently, it is well adapted to implementation on a parallel architecture such as the graphics processing unit. 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 runs on a ATI Mobility Radeon HD 3650 GPU. It 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.

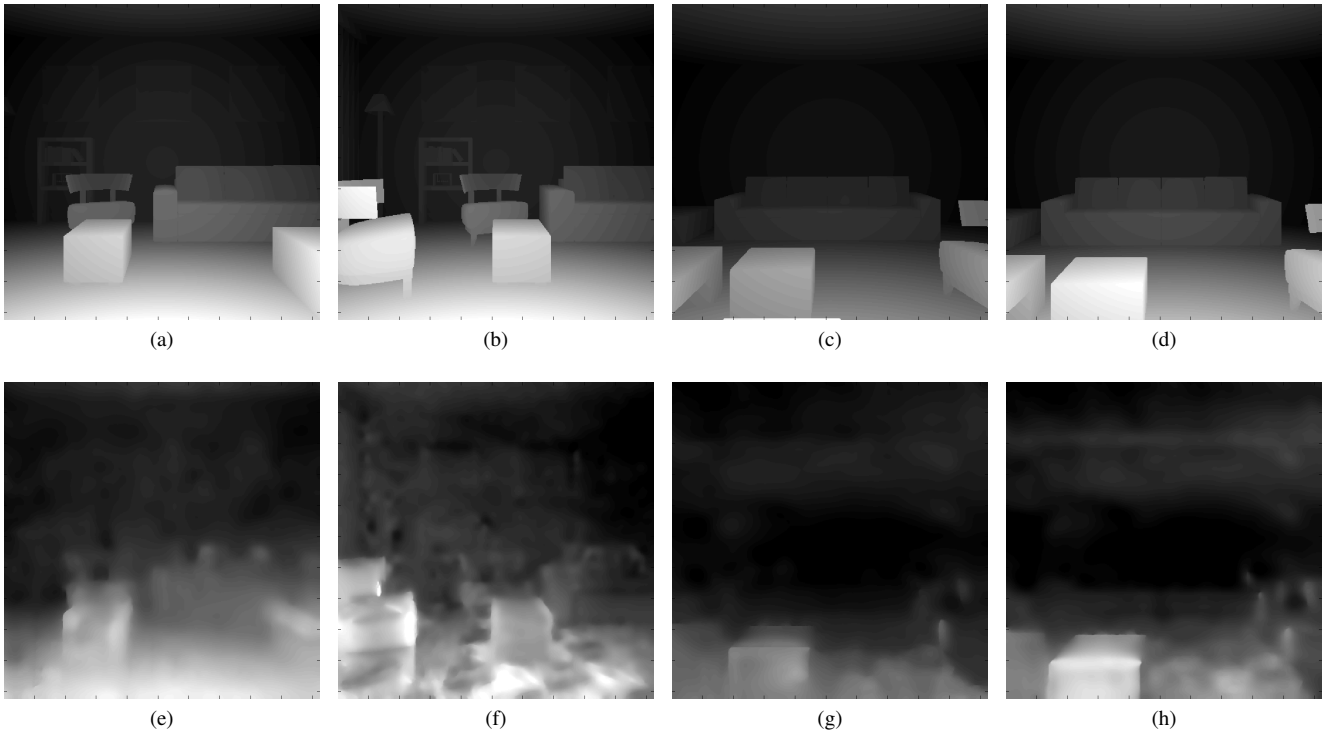

Figure 5: Depth maps. Top row: ground truth depth maps. Bottom row: recovered depth maps with the *joint* ego-motion and depth estimation algorithm. 1st and 2nd column: movement in x direction. 3rd and 4th column: movement in z direction. MSEs of the estimated depth $d(\mathbf{r})$: e) 3.9 , f) 7.4 , g) 6.6 and h) 6.1. MSEs of the estimated depth map $Z(\mathbf{r})$: e) $6.2 \cdot 10^{-4}$, f) $10.1 \cdot 10^{-4}$, g) $4.1 \cdot 10^{-4}$ and h) $3.9 \cdot 10^{-4}$.

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

[1] L. Bagnato, P. Frossard, and P. Vandergheynst. Optical flow and depth from motion for omnidirectional images using a tv-l1 variational framework on graphs. In *Proceedings of ICIP*, Cairo, Egypt, 2009.

[2] L. Bagnato, P. Vandergheynst, and P. Frossard. A Variational Framework for Structure from Motion in Omnidirectional Image Sequences. Technical report, EPFL, 2009.

[3] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97, 2004.

[4] O. Faugeras and Q.-T. Luong. *The Geometry of Multiple Images. The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. The MIT Press, 2001.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[6] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[7] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, 2006.

[8] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. pages 1–8, 2007.

[9] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *DAGM-Symposium*, pages 214–223, 2007.

[10] G. Zhang, J. Jia, T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31: 974–988, 2009.