

## Journal Pre-proof

Joint direct estimation of 3D geometry and 3D motion using spatio temporal gradients

Francisco Barranco, Cornelia Fermüller, Yiannis Aloimonos, Eduardo Ros

PII: S0031-3203(20)30562-8  
DOI: <https://doi.org/10.1016/j.patcog.2020.107759>  
Reference: PR 107759



To appear in: *Pattern Recognition*

Received date: 13 May 2018  
Revised date: 14 November 2020  
Accepted date: 23 November 2020

Please cite this article as: Francisco Barranco, Cornelia Fermüller, Yiannis Aloimonos, Eduardo Ros, Joint direct estimation of 3D geometry and 3D motion using spatio temporal gradients, *Pattern Recognition* (2020), doi: <https://doi.org/10.1016/j.patcog.2020.107759>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier Ltd.

**Highlights**

- This method does not require optical flow computation, it works on normal flow
- Normal flow is derived only from spatial and temporal gradients
- Using normal flow prevents from accumulating error from assumptions
- We propose a non-convex optimization and the positive depth constraint
- We refine through linear optimization via the 3D structure estimation

# Joint direct estimation of 3D geometry and 3D motion using spatio temporal gradients

Francisco Barranco<sup>a,\*</sup>, Cornelia Fermüller<sup>b</sup>, Yiannis Aloimonos<sup>b</sup>,  
Eduardo Ros<sup>a</sup>

<sup>a</sup>*Dept. of Computer Architecture and Technology, CITIC, University of Granada, Spain.*

<sup>b</sup>*Dept. of Computer Science, UMIACS, University of Maryland, College Park, MD, USA.*

---

## Abstract

Conventional image-motion based methods for structure from motion first compute optical flow, then solve for the 3D motion parameters based on the epipolar constraint, and finally recover the 3D geometry of the scene. However, errors in optical flow due to regularization can lead to large errors in 3D motion and structure. This paper investigates whether performance and consistency can be improved by avoiding optical flow estimation in the early stages of the structure-from-motion pipeline, and it proposes a new direct method based on image gradients (normal flow) only. Our main idea lies in a reformulation of the positive-depth constraint – the basis for estimating egomotion from normal flow – as a continuous piecewise differentiable function, which allows the use of well-known minimization techniques to solve for 3D motion. The 3D motion estimate is then refined and structure estimated adding a regularization based on depth. Experimental comparisons on standard synthetic datasets and the real-world driving benchmark dataset Kitti using three different optic flow algorithms show that the method achieves better accuracy in all but one case. Furthermore, it outperforms existing normal flow based 3D motion estimation techniques. Finally, the recovered 3D geometry is shown to be also very accurate.

**Keywords:** 3D motion, egomotion, structure from motion, normal flow.

---

\*Corresponding author

Email address: `fbarranco@ugr.es` (Francisco Barranco)

## 1. Introduction

The problem of egomotion or self-motion estimation from a moving monocular observer, after many years of research, is still considered a difficult problem. Recently it has attracted renewed attention in the Computer Vision community due to emerging applications in robotics, autonomous navigation and augmented reality. Physically, the motion of the camera can be interpreted as the linear combination of a 3D translation followed by a 3D rotation. The instantaneous motion captured contains information about the camera's 3D motion and the 3D scene geometry. Egomotion estimation amounts to computing five parameters: three for the 3D rotation and two for the axis of the 3D translation, because without additional information, there is an ambiguity between translational velocity and depth. Based on the 3D motion, the 3D relative structure can be estimated.

The classic approach to estimating structure and motion employs three steps: first, the full dense optical flow between successive frames is estimated; second, the 3D translation and 3D rotation are recovered using the optical flow, possibly making assumptions about the camera motion and the scene; third, the 3D geometry up to the scaling factor is estimated [1, 2, 3, 4]. Instead of dense flow, a sparse set of feature correspondences is often used, as is common in standard visual odometry and SLAM methods [5, 6, 7]. However, recent SLAM formulations [8] do not estimate 3D motion through constraints independent of depth, but estimate 3D motion and depth combined by minimizing photometric/geometric distance that explain image patch matches. The focus of this paper is on the evaluation of depth independent constraints.

The main constraint to estimate 3D motion from video independent of structure, is the epipolar constraint. However, it requires as input optical flow or correspondences. One problem is that optical flow cannot be estimated accurately. Most top optical flow techniques are based on the work of Horn & Schunk [9]. The key assumption is that the change of the intensity over a small time

interval remains constant. Since this only provides one equation and flow fields are two-dimensional, additional constraints on the flow field are enforced assuming a smooth variation of the field spatially in local neighborhoods [10, 11, 12]. These assumptions cause the optical flow to be imprecise at object contours, where there are occlusions or when the motion is large. Motion fields do not vary smoothly close to object boundaries, occlusions cause mismatches, and large motions violate the assumption of local intensity constancy.

Another motion constraint, independent of structure, is the depth positivity constraint [13, 14, 15, 16, 17] also referred to as cheirality constraint [18]. The scene has to be in front of the camera, and thus the depth has to be positive. This constraint can be used directly from "*measurable image quantities*", i.e. the spatial and temporal image intensity gradients, or normal flow, in so-called direct methods. The positivity constraint is the only constraint applicable to normal flow without making assumptions on scene depth or shape. Although, it can also be applied to optical flow. It has not been popular, because there has not been a formulation allowing to implement the depth positivity inequality in an efficient way. This work proposes such a formulation of depth positivity, borrowing from current machine-learning models. On its basis then a new direct method for estimation of 3D motion and structure is proposed.

The main contributions of this paper are: 1) a new formulation of the depth positivity constraint that allows for efficient minimization using a negative ReLU function; 2) a new direct method for egomotion and 3D structure estimation from normal flow only, that starts with estimating 3D motion from the depth positivity constraint, then uses a regularization to obtain depth, and finally refines both estimates; 3) a comparison of the method to a number of studies using optical flow and to other direct approaches using normal flow on a number of datasets including real-world sequences, which show that the method outperforms previous methods using different error metrics.

## 2. Related methods

Longuet-Higgins in his seminal work presented the constraints for computing egomotion from image measurements [19]. Classic approaches that use optical flow model the 3D rigid displacement between frames with instantaneous 3D velocity [20]. Bruss and Horn [21] used the weighted bilinear constraint and solved for the egomotion using a least-squares optimization. Kanatani [22] proposed a method based on the differential epipolar constraints that accounts and corrects for the bias in the translational velocity. Modern optimal methods compute more accurate solutions at the expense of nonlinear objective functions and the risk of falling into local minima [3]. Zhang [4] considered a two-step iterative approach that estimates the translation separately from the rotation, using numerical Gauss-Newton approximation. Pauwels and van Halle [3] proposed a method that integrates depth cues into the minimization of the squared-distance image reprojection error. This method improved previous works by reducing the risk of falling into local minima. Instead of using random initializations or initializations from classical linear method solutions, the authors used a weighting strategy to change the complexity at each iteration, making the algorithm more robust. Raudies and Neumann [1, 2] also used a polynomial solution for the bilinear constraint employing auxiliary variables (as done in [22]) for a more efficient method of linear complexity; it applies RANSAC to reduce the impact of noise as well. For a complete review on 3D motion estimation methods see [2].

On the other hand, many methods are feature-based. During the last years in the context of autonomous navigation, most 3D motion approaches proposed are SLAM methods. The most recent work from Engel *et al.*[8] presented a real-time method for visual odometry that jointly estimates egomotion and geometry without using a smoothing prior, but by sampling a set of locations throughout the sequence. The same authors developed LSD-SLAM [23], which uses a photometric error measure and a geometric smoothing prior to solve the optimization on a full dense optical flow field throughout consecutive frames.

Mur-Artal *et al.*[6] presented an open-source framework that allows for odometry estimation, using a geometric distance measure without geometric prior.

90 Ranftl *et al.*[24] proposed a monocular depth estimation approach that jointly solves for egomotion using a full dense regularized optical flow field assuming its smoothness. Finally, Forster *et al.*[7] also described a probabilistic hybrid approach for semi-direct visual odometry for UAVs, that uses a formulation based on a geometric prior for the initialization but avoids its use in the subsequent

95 model optimization.

Several works in the literature have studied direct methods that use normal flow avoiding optical flow estimation. Aloimonos and Brown [25] studied the case of rotation, Horn and Weldon [13], Neghadaripour and Horn [14] and Sinclair *et al.*[26] proposed normal flow based methods for translation and stud-

100 ied the robustness, and Carlsson [15] studied constraints on scene structure for unique 3D motion estimation. Fermüller and Aloimonos developed methods [17, 27] for the general case, that formulate 3D motion estimation as a pattern recognition problem. Subsets of normal flow vectors define global patterns in the image plane whose location encodes the egomotion parameters. Finally, Brod-

105 sky *et al.*[28] estimated 3D motion from normal flow assuming planar patches, and Ji and Fermüller based on this method proposed a solution [29] that combines the estimated motion fields from consecutive frames using a constraints on the inverse depth for regular image patches and for segmented regions.

More recently, Hui and Chung [30] used a binocular sequence but computed

110 monocular motion with normal flows and then estimated depth from the stereo system without explicitly computing correspondences. They optimized for the 3D motion using the so-called Apparent Flow Direction (AFD) and Apparent Flow Magnitude (AFM) constraints. The first constraint, AFD, is a relaxation of the positive depth constraint. The second constraint, AFM, affects the mo-

115 tion magnitude and only uses the normal flows orthogonal to the translational component. The same authors in [31] presented a more comprehensive evaluation and compared their results with optical flow methods. In [32] the authors described a two-step method that first estimates the 3D motion using only the

normal flow vectors orthogonal to the translational motion component. Next  
 120 the method solves for the rotation discarding the solutions that lie out of the  
 half-plane consistent with the normal flow estimates, defined by the first step.  
 In [33] the same authors utilize k-means clustering to group the flow vectors  
 that support the same FOE candidate. Next, the rotational parameters are  
 estimated using RANSAC and finally, a confirmation strategy based on the  
 125 half-plane constraint helps finding potential solutions. Finally, using the same  
 normal flow constraints, other works extract the 3D structure from the scene  
 [34, 30].

### 3. Egomotion estimation from normal flow

Let us first rephrase the egomotion problem definition as follows: recover the  
 130 3D trajectory and pose of a monocular observer undergoing a rigid motion in a  
 stationary environment. Assuming that the coordinate system of the camera is  
 centered in the principal point and  $f$  is the focal length, the 3D velocity  $\mathbf{v}$  of a  
 point  $\mathbf{x}$  is defined as  $\mathbf{v} = -\mathbf{t} - \mathbf{w} \times \mathbf{x}$ , where  $\mathbf{t} = (t_x, t_y, t_z)^T$  is the velocity of  
 translation,  $\mathbf{w} = (w_x, w_y, w_z)^T$  is the velocity of rotation, and  $\times$  represents the  
 135 cross product. Then, the motion field  $\mathbf{u}(\mathbf{x}) = (u, v)^T$  at location  $\mathbf{x} = (x, y)^T$  is  
 related to the 3D velocities [35] as

$$\mathbf{u}(\mathbf{x}) = \frac{1}{Z(\mathbf{x})} A(\mathbf{x}) \mathbf{t} + B(\mathbf{x}) \mathbf{w} \quad (1)$$

with

$$A(\mathbf{x}) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}, \quad (2)$$

$$B(\mathbf{x}) = \begin{bmatrix} \frac{xy}{f} & \left(-\frac{x^2}{f} - f\right) & y \\ \left(\frac{y^2}{f} + f\right) & -\frac{xy}{f} & -x \end{bmatrix}$$

where the motion is expressed as the sum of the rotational component  $B(\mathbf{x})\mathbf{w}$ ,  
 and the translational component  $\frac{1}{Z(\mathbf{x})}A(\mathbf{x})\mathbf{t}$ , which also depends on the depth of  
 the scene  $Z$ . Due to this dependency there are five parameters to be estimated,

140 namely the translational velocity axis  $(t_x/t_z, t_y/t_z)^T$  and the rotational velocity  $(w_x, w_y, w_z)^T$ .

### 3.1. Normal flow

To estimate motion fields, we use the so-called *optical flow constraint* (see [9]), which assumes that the intensity  $I$  at a point remains constant over a short time interval  $\delta t$ . Approximating the image brightness function with a first order Taylor expansion, we obtain Eq. (3)

$$0 = I(x, y, t) - I(x + u\delta t, y + v\delta t, t + \delta t) \approx I_x u + I_y v + I_t, \quad (3)$$

where  $I_x, I_y, I_t$  are the partial derivatives of the brightness, and  $\mathbf{u} = (u, v)$  is the image motion. Equation 3 provides one constraint and defines the flow component parallel to the gradient. To obtain a second component, additional  
145 assumptions, such as smoothness of the flow or a parametric model in the image coordinates are assumed.

Direct methods do not require computing the two-dimensional optical flow, but instead use directly the image gradients as input (equation 3). For a geometric interpretation, we consider the projection of the optical flow on the gradient direction, called the *normal flow*, which amounts to:

$$\mathbf{u}_n(\mathbf{x}) = \frac{-I_t(\mathbf{x})}{\|\nabla I(\mathbf{x})\|^2} \nabla I(\mathbf{x}) \quad (4)$$

where  $\nabla I = (I_x, I_y)$  is the intensity spatial gradient. Given the gradient direction as a unitary vector  $\mathbf{n} = (n_x, n_y)$ , the normal flow speed amounts to  
150  $|\mathbf{u}_n(\mathbf{x})| = \mathbf{n}(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x})$  where  $\cdot$  represents the inner product (see also [17]). From now on, we use  $u_n(\mathbf{x})$  instead of  $|\mathbf{u}_n(\mathbf{x})|$  in order to simplify the notation. Considering the normal flow instead of the optical flow, we obtain from Eq. (1) by multiplying both sides with  $\mathbf{n} = (n_x, n_y)$  :

$$u_n(\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x}) \mathbf{t} + \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x}) \mathbf{w} \quad (5)$$

### 3.2. Constraints for the objective function

155 The 3D motion parameters and the depth are estimated using various optimization constraints. A first approximation minimizes the squared distances with respect to the model as in Eq. (6)

$$\operatorname{argmin}_{\mathbf{t}, \mathbf{w}} \sum_{i=1}^N \left\| \mathbf{u}(\mathbf{x}_i) - \frac{1}{Z_i} A(\mathbf{x}_i) \mathbf{t} - B(\mathbf{x}_i) \mathbf{w} \right\|_2, \quad (6)$$

where  $\mathbf{x}_i$  is a position of the image  $(x_i, y_i)$ , and  $N$  is the total number of flow estimates of the image. Thus, the system has  $N$  constraints where we have  $N$  depth unknowns  $Z_i$  and the parameters for  $\mathbf{t}$  and  $\mathbf{w}$ .  
160

The classic approach removes the depth from Eq. (6), obtaining the so-called *epipolar constraint* [21]. It can be written in the form:

$$\operatorname{argmin}_{\mathbf{t}, \mathbf{w}} \sum_{i=1}^N \left\| (\mathbf{u}(\mathbf{x}_i) - B(\mathbf{x}_i) \mathbf{w})^T \cdot (A(\mathbf{x}_i) \mathbf{t})_{\perp} \right\|_2, \quad (7)$$

where  $(A(\mathbf{x}_i) \mathbf{t})_{\perp}$  denotes that the vector is orthogonal to the vector  $A(\mathbf{x}_i) \mathbf{t}$ . This constraint is also referred in the literature simply as the bilinear constraint, as it is linear in the translation, or the rotation. Optimizing this unweighted bilinear constraint introduces a statistical bias [36]. This is avoided with a slightly modified version, that includes a weighing factor (for example, with  $\rho = 1$ )  
165

$$\operatorname{argmin}_{\mathbf{t}, \mathbf{w}} \sum_{i=1}^N \left\| (\mathbf{u}(\mathbf{x}_i) - B(\mathbf{x}_i) \mathbf{w})^T \cdot \frac{(A(\mathbf{x}_i) \mathbf{t})_{\perp}}{(\|A(\mathbf{x}_i) \mathbf{t}\|)^{\rho}} \right\|_2. \quad (8)$$

Many methods solve the bilinear constraint or its weighted variations (e.g. Eq. (8)) in a 2-step approach: they first search for either the rotation or the translation and then solve for the other one. Other formulations, similar to the use of the essential matrix in the discrete case, solve first for intermediate parameters that encode both rotation and translation [37].  
170

Another approach is to employ a scene model [38, 39]. In [40, 29] solutions were proposed that assume 3D planar piece-wise depth. Let us assume  $P$  patches where  $P \ll N$ ; if all points  $(x_k, y_k)$  in the image patch  $p$  lie on the same plane

$\Gamma_p$  then:

$$\Gamma_p = \alpha_p \frac{x_k}{f} + \beta_p \frac{y_k}{f} + \gamma_p = \frac{1}{Z_k}, \quad (9)$$

where  $\mathbf{v}_p = (\alpha_p, \beta_p, \gamma_p)$  stands for the plane vector and  $f$  is the focal length.

As mentioned before, in order to compute the 3D motion and relative depth, only the axis of the translational motion is required. The equation can be rewritten using  $C_k = \|\mathbf{t}\| (Z_k)^{-1}$  to denote the scaled inverse depth, and  $\tilde{\mathbf{t}} = \mathbf{t} / \|\mathbf{t}\|$  to denote the translation axis. Due to this change, instead of optimizing for plane parameters  $\mathbf{v}_p$ , we optimize for  $\tilde{\mathbf{v}}_p = \|\mathbf{t}\| (\alpha_p, \beta_p, \gamma_p)$ . We then obtain an overdetermined system with  $N$  equations for a total number of  $3 * P + 5$  unknowns. Denoting individual patches as  $p_i$ , each with  $K_{p_i}$  image motion measurements, we have:

$$\arg \min_{\tilde{\mathbf{t}}, \mathbf{w}, \tilde{\mathbf{v}}_{p_i}} \sum_{j=1}^P \sum_{k=1}^{K_{p_i}} \|\mathbf{u}(\mathbf{x}_k) - \Gamma_{p_i}(\tilde{\mathbf{v}}_{p_i}, \mathbf{x}_k) A(\mathbf{x}_k) \tilde{\mathbf{t}} - B(\mathbf{x}_k) \mathbf{w}\|_2. \quad (10)$$

We can apply the same approach to normal flow. If our measurements are the normal flow speeds  $u_n(\mathbf{x}_k) = \mathbf{n}(\mathbf{x}_k) \cdot \mathbf{u}(\mathbf{x}_k)$ , we obtain from equation 10

$$\arg \min_{\tilde{\mathbf{t}}, \mathbf{w}, \tilde{\mathbf{v}}_{p_i}} \sum_{j=1}^P \sum_{k=1}^{K_{p_i}} \|u_n(\mathbf{x}_k) - \mathbf{n}(\mathbf{x}_k) \cdot \Gamma_{p_i}(\tilde{\mathbf{v}}_{p_i}, \mathbf{x}_k) A(\mathbf{x}_k) \tilde{\mathbf{t}} - \mathbf{n}(\mathbf{x}_k) \cdot B(\mathbf{x}_k) \mathbf{w}\|_2. \quad (11)$$

We have implemented for comparison a version of this method that uses regular patches to partition the image. One could first segment the scene to obtain the patches, which however as shown in [29] does not lead to significant accuracy improvement for the 3D motion estimation.

The above constraints require assumptions on the local smoothness of image motion, either for computing optical flow that is used in the epipolar constraint or indirectly in the assumption of planar scene patches. However, there is another weaker constraint, that can be used with normal flow only, thus avoiding these assumptions. This is the so-called positive depth constraint. Since the scene in view is in front of the camera, its depth is positive, i.e.  $Z > 0$ . From Eq. (5), subtracting the rotational component, we can derive that the derotated normal flow and the translational flow need to have same sign for  $Z$  to be

positive. Thus

$$(u_n(\mathbf{x}) - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}}) > 0. \quad (12)$$

This constraint can be optimized with a voting function that counts the number of negative depth values, as discussed in the stability analysis in [41].

200 One can search for the  $\tilde{\mathbf{t}}$  and  $\mathbf{w}$  with the smallest number of normal flow values that make Eq. (12) negative. This minimization can be expressed as

$$\arg \min_{\tilde{\mathbf{t}}, \mathbf{w}} \sum_{i=1}^N \mathcal{V}(\mathbf{x}_i, \tilde{\mathbf{t}}, \mathbf{w}) \quad \text{with} \quad (13)$$

$$\begin{aligned} \mathcal{V}(\mathbf{x}, \tilde{\mathbf{t}}, \mathbf{w}) = & \\ \begin{cases} 0 & \text{if } (u_n(\mathbf{x}) - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}}) > 0 \\ 1 & \text{if } (u_n(\mathbf{x}) - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}}) < 0 \end{cases} \end{aligned} \quad (14)$$

An algorithmic solution, which only uses the sign of the normal flow was given in [42]. By considering subsets of normal flow values in selected directions, the axis of translation and axis of rotation can be derived using a pattern matching approach in lower dimensional spaces. However, the proposed constraint – a relaxed version of the voting function in Eq. (14) – is weaker and if  
205 used by itself can only provide bounds for the solution.

$$\begin{aligned} \mathcal{V}_r(\mathbf{x}, \tilde{\mathbf{t}}, \mathbf{w}) = & \\ \begin{cases} 1 & \text{if } u_n(\mathbf{x}) > 0, \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w} < 0, \mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}} < 0 \\ 1 & \text{if } u_n(\mathbf{x}) < 0, \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w} > 0, \mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}} > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (15)$$

There are variations that ease the computation by removing either the rotational or translational motion from the previous constraint. The first variation  
210 considers that the rotational motion can be computed accurately using inertial sensors (see [43]). In this case, the rotational motion component is subtracted from the normal flow speed, and one has to solve the minimization only for

the translation. In practice, an additional step of sensor fusion to combine the inertial and visual data is required for this solution, because the sensors have  
 215 different noise models and rates.

The second variation heavily simplifies the minimization. Given a translation, this constraint only considers the normal flow vectors orthogonal to the translational motion component. For these components the translational motion term is zero, i.e.  $A(\mathbf{x})\tilde{\mathbf{t}} \cdot \mathbf{n} = 0$ . In this case, one searches for the translational  
 220 axis  $\tilde{\mathbf{t}}$ , and for each axis solves an optimization in the rotational motion as

$$\arg \min_{\mathbf{w}} \sum_{j=1}^M \|u_n(\mathbf{x}_j) - \mathbf{n}(\mathbf{x}_j) \cdot B(\mathbf{x}_j)\mathbf{w}\|, \quad (16)$$

where  $M$  is the number of points  $\mathbf{x}_j$  with normal flows orthogonal to the translation motion components and thus  $M \leq N$ . The main drawback here is that the number of points with translational flow perpendicular to the gradient can be much smaller than  $N$ , and thus the estimation becomes less accurate. Some  
 225 works combined this with additional constraints to achieve a solution for the general motion case [31].

#### 4. Our direct approach using normal flow

The method proceeds in three steps: First, based on a new formulation of the depth positivity constraint that leads to a convex optimization, 3D motion  
 230 is estimated using an interior point method. The solution is found by first searching for the translation axis, and then optimizing successively for the rotation and the translation. Second, using the estimate for the egomotion, 3D structure is computed and a regularization is imposed on the structure using an inpainting technique. Thirdly, the egomotion and structure are iteratively  
 235 refined. A convergence threshold stops the iterative process.

##### 4.1. Egomotion estimation using normal flow

Let us denote the left hand side of the positive depth constraint in Eq. (12) as  $f(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}) = (u_n(\mathbf{x}) - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}})$ . We then model the

inequality in Eq. (12) using the negative ReLu function, which we denote as  $\mathcal{H}$ ,  
 240 and reformulate the optimization problem in Eq. (13) as

$$\arg \min_{\tilde{\mathbf{t}}, \mathbf{w}} \sum_{i=1}^N \mathcal{H}(f(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}_i)) \quad (17)$$

where

$$\mathcal{H}(x) = \begin{cases} -x & \text{if } x \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

To solve for the 3D motion, we iteratively solve for  $\tilde{\mathbf{t}}$  and  $\mathbf{w}$ . The method starts by searching for the translation axis. Given a candidate translation, we optimize the objective function  $\sum_{i=1}^N \mathcal{H}(f(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}_i))$  to solve for the rotation  $\mathbf{w}$ . With this rotation, the translation is re-estimated substituting in Eq. (17) to  
 245 obtain a more accurate solution for the translation axis.

In our implementation, we optimize with an interior point method, and thus we only need to provide the objective function and its gradient. However  $\mathcal{H}(x)$  is not strong convex since  $\mathcal{H}''(x) = 0$ , and thus it cannot be used in Newton-type optimization methods. Since the gradient  $\mathcal{H}'(x)$  has a singular point at 0, we use the following smooth approximation, as is common in machine learning:

$$\mathcal{H}'(x) = \begin{cases} -1 & \text{if } x \leq -\epsilon \\ -1/2 & \text{if } -\epsilon < x < \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

where the gradient is defined as

$$\frac{\partial \sum_{i=1}^N \mathcal{H}(f(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}_i))}{\partial \mathbf{w}} = \sum_{i=1}^N \mathcal{H}'(f(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}_i)) \cdot f'(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}_i) \quad (20)$$

with

$$f'(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}) = (\mathbf{n}(\mathbf{x})A(\mathbf{x})\tilde{\mathbf{t}}) \cdot (B(\mathbf{x})^T (-\mathbf{n}(\mathbf{x})^T)). \quad (21)$$

In our implementation, we use the interior-point method described in [44] and reformulated in [45] to solve the optimization. For the initialization we use  $\mathbf{w} = [0, 0, 0]^T$  and  $\tilde{\mathbf{t}} = [0, 0]^T$ . The stopping criteria is based on the tolerance

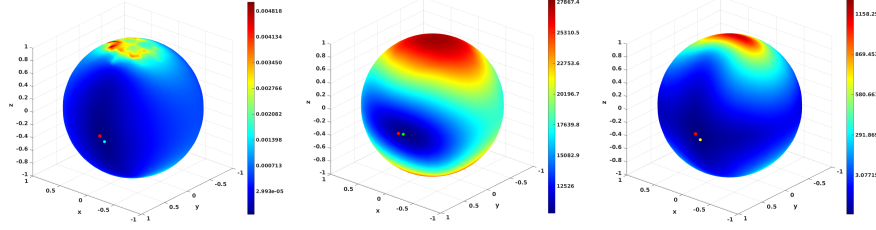


Figure 1: Illustrative example of residual error surfaces for a randomly generated 3D motion, shown in the 2D subspace mapped onto a sphere surface that represents the translation direction. From left to right, top row: a) bilinear constraint from Eq. (8), b) epipolar constraint assuming planar patches in Eq. (10), c) positive depth constraint with normal flow vectors in Eq. (12). The ground-truth is marked with a red dot and the estimated solution with a colored smaller dot.

between consecutive steps (set to  $1e-15$ ) and the tolerance on the change of the value of the objective function between consecutive steps (also set to  $1e-15$ ).

Next, we visualize the behavior of the different minimization functions. Fig. 1 shows the residuals of the optimization in the 2D subspace of translation directions on the surface of the sphere. For this illustrative example, we chose a random 3D motion and randomly generated depth values. For each possible translation, the residual is computed by solving for the optimal rotation. The ground-truth is marked with a red dot and the results from the different optimizations are marked with smaller dots in different colors.

Sphere *a* shows the optimization for the bilinear constraint in Eq. (8). Sphere *b* shows the optimization of the constraint assuming planar patches as in Eq. (10). Finally, sphere *c* shows the results for the optimization using the positive depth constraint as in Eq. (12).

Sphere *b* shows a smaller region of low-valued residuals compared to the other two, but the optimization is done for the motion and planar shape parameters of all patches used. A small number of patches violates the initial hypothesis that assumes all points in the patch to lie on the same plane, and more patches increase the number of unknowns, increasing also the complexity of the minimization. The surfaces due to the the epipolar constraint in *a* and the depth

positive constraint  $c$  show a similar minimum error region, although Sphere  $c$  has a smoother surface, and in this case, a smaller error between the computed solution and the ground-truth. However, for all the residuals shown here, the optical and normal flow values are the ground-truth, while real applications are affected by noise.

#### 4.2. Refining positive depth constraint

In order to refine the egomotion solution proposed in §4.1, we use the 3D structure from the translational motion. A second step is added estimating first the 3D structure as follows

$$C(\mathbf{x}) = Z(\mathbf{x}) / \|\mathbf{t}\| = \frac{A(\mathbf{x})\hat{\mathbf{t}} \cdot \mathbf{n}(\mathbf{x})}{u_n(\mathbf{x}) - B(\mathbf{x})\hat{\mathbf{w}} \cdot \mathbf{n}(\mathbf{x})} \quad (22)$$

where  $C$  is the structure and represents the inverse depth up to a factor. Here  $\hat{\mathbf{t}}$  represents the translation axis and  $\hat{\mathbf{w}}$  the rotational velocity estimated with the previous method. After that, a regularization process is performed on the structure using the inpainting method in [46]. This method has been designed for reconstructing parts of depth maps that are lost e.g. when using infrared-based Kinect sensors. In our case, the estimation is sparse since normal flow is only computed at locations of large image gradients such as at edges or object contours. The inpainting method reconstructs the depth at smooth surfaces while regularizing the estimates at object contours. It uses a second-order smoothness assumption as is common in natural images (see [47]). However, this assumption is violated close to object contours due to the depth discontinuities.

After the depth regularization, a simpler least-squared minimization can be performed to obtain refined motion estimates optimizing

$$\arg \min_{\tilde{\mathbf{t}}, \mathbf{w}} \sum_{i=1}^N \|u_n(\mathbf{x}_i) - (C(\mathbf{x}_i)A(\mathbf{x}_i)\tilde{\mathbf{t}} - B(\mathbf{x}_i)\mathbf{w}) \cdot \mathbf{n}(\mathbf{x}_i)\|_2. \quad (23)$$

In our approach, these two steps are iteratively repeated further refining the solutions for the 3D motion and the 3D structure. The stopping criterion here is the sum of differences of the 3D structure between consecutive estimates, for each point (convergence threshold).

## 5. Experiments

295 This section presents: 1) an evaluation of our direct method using different metrics on various datasets, 2) a comparison of our direct approach with methods that use optical flow and with other direct methods that use normal flow, and 3) an evaluation of our method recovery of 3D structure.

We evaluate the ego-motion estimation on the following four datasets:

- 300 1. The Artificial dataset contains 5000 random sequences of images of size 150 x 150 with a field of view (FOV) of 30°, and an image plane of dimension 0.01 x 0.01 meters. The rotational velocity is up to 20° per frame, the translational velocity is up to 3 meters per frame, and the maximum virtual depth of the scene is 10 meters.
- 305 2. The Yosemite sequence [48] has images of size 316 x 252, and is a simulation of a fly-through the Yosemite Valley with divergent motion and translational motion in the clouds. However, the clouds, as is common, have been masked out. The translation is  $\mathbf{t} = [0, 0.17, 0.98]^T$  with a speed of 34.8 pixels per frame, the rotational motion is  $\mathbf{w} = [0.0133, 0.0931, 0.0162]^T$  in degrees per frame, and the FOV is 40°.
- 310 3. The Fountain sequence [49] is a synthetic sequence with images of size 320 x 240 of a curvilinear motion featuring a patio sequence surrounded by columns and a central fountain. The ground-truth optical flow, 3D pose, 3D velocity, and depth are provided. The translation is  $\mathbf{t} = [-0.2578, 0.0872, 0.9622]^T$  with a speed of 2.5 pixels per frame, the rotational motion is  $\mathbf{w} = [-0.125, 0.20, -0.125]^T$  degrees per frame, and the FOV is 40°.
- 315 4. The Kitti dataset [50] for visual odometry is a popular set of 22 driving sequences of stereo road scenes. These are long natural scenes with different trajectory directions and speeds. To the best of our knowledge, our study is the first to evaluate a direct 3D motion estimation method on such a complex real-world dataset. We have used 11 of the sequences for which the ground-truth is provided: the total number of frames is more than 22500. Since there is very large inter-frame displacement up to 50
- 320

pixels, making it impossible to estimate accurate intensity gradient, we  
 325 interpolate frames to reduce the maximum displacement to 3 to 5 pixels.

A preprocessing stage is performed when using our direct method for all  
 datasets. We use a 5 x 5 Gaussian filter with  $\sigma = 1.05$ . For the normal flow  
 estimation, we use a 2D derivative filter with kernel  $f^T * f$  with  $f = [-1, 9, -$   
 45, 0, 45, -9, 1]/60 for the spatial derivatives, and kernel  $[-1, 1]/2$  for the  
 330 temporal derivative.

For the evaluation of 3D motion estimation we use two error metrics: the  
 average angular error (AAE) which is defined as the average angular distance  
 between the ground-truth  $\mathbf{u}$  and the estimated motion vector  $\hat{\mathbf{u}}$  as

$$AAE = \frac{1}{N} \sum_{i=1}^N \arccos \left( \frac{\hat{\mathbf{u}}(\mathbf{x}_i)^T \mathbf{u}(\mathbf{x}_i)}{\|\hat{\mathbf{u}}(\mathbf{x}_i)\| \|\mathbf{u}(\mathbf{x}_i)\|} \right), \quad (24)$$

and the EPE, which is the average euclidean distance between the ground-truth  
 and the estimated motion vector, defined as

$$EPE = \frac{1}{N} \sum_{i=1}^N \|\mathbf{u}(\mathbf{x}_i) - \hat{\mathbf{u}}(\mathbf{x}_i)\|. \quad (25)$$

Since we are estimating only the translation direction, we evaluate the AAE  
 and EPE for rotation, but only the AAE for translation. For the Kitti dataset  
 we do not provide the rotational AAE, because the rotational direction varies  
 largely making it difficult to interpret this error.

We also use two error metrics to evaluate the estimated structure: the mean  
 absolute error (MAE), defined as the average absolute difference between the es-  
 timated 3D structure (times the translational speed)  $\hat{d}$  and the disparity ground-  
 truth  $d$

$$MAE = \frac{1}{N} \sum_{i=1}^N |d(\mathbf{x}_i) - \hat{d}(\mathbf{x}_i)|, \quad (26)$$

335 and the PoBP, defined as the percentage of points with a MAE greater than 1  
 (see [51]). This error gives a measure of the accuracy of the recovered structure  
 at object contours.

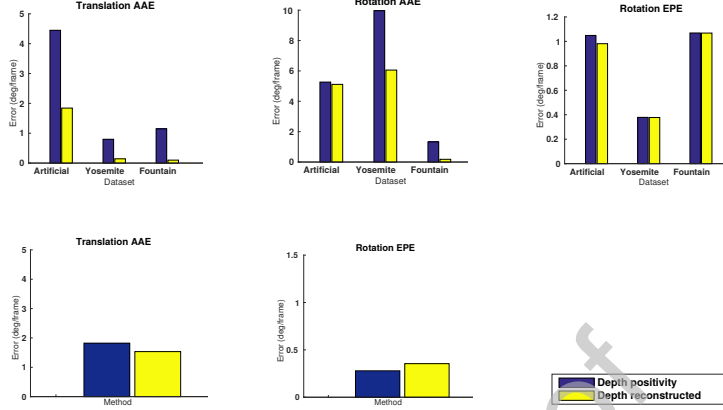


Figure 2: Error plots for 3D motion estimation estimated using our proposed method. The first row shows the translation AAE and rotation AAE and EPE for the Artificial, Yosemite, and Fountain datasets. The second row shows the translation AAE and the rotation EPE for the Kitti dataset, averaging over all frames and sequences. For the artificial dataset, the normal flow is estimated in a random direction using the ground-truth; for the other datasets, the normal flow is computed using the spatio-temporal gradients. The blue and yellow bars show the error when using only the refined positive depth constraint, and after normalizing with the 3D structure reconstruction.

### 5.1. Our direct approach for 3D motion estimation

The plots in Fig. 2 illustrate the results of 3D motion estimation discussed in Section §4 for a) the depth-independent minimization using the depth positivity constraint, and b) after inclusion of the refinement step based on the depth map. The first row shows the translational AAE and the rotational AAE and EPE for the Artificial, Yosemite, and Fountain datasets. After the 3D reconstruction, the error is reduced substantially for all datasets. For the Kitti dataset, the average error reduction is 10% to 15% for the translational AAE. The rotational EPE increases, but this increase is negligible for the given driving scenario, where the rotation is very small.

Fig. 3 shows for the Kitti benchmark the estimated paths projected on the X-Z plane. For the scale, we used the ground-truth translation speed. For each case, we show the trajectories for all the methods and the ground-truth. Most

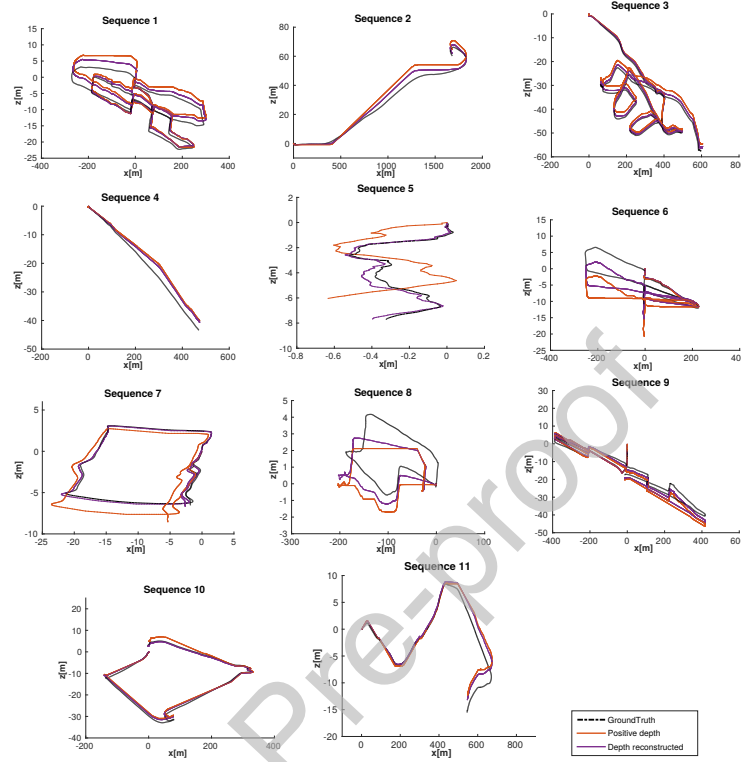


Figure 3: Estimated paths projected onto the X-Z plane computed using our proposed method for the 11 sequences of the Kitti dataset. Results for [Sequence 6 and 8](#) show more distortion but, in general, the trajectories are successfully recovered, and the method that normalizes the estimation through the reconstruction of the 3D geometry of the scene achieves the best results.

estimated trajectories are very similar to the ground-truth, and particularly after 3D reconstruction there is very high accuracy. However, for [Sequence 5](#) only the refined solution reaches a successful trajectory. This could be due to the small translational motion with respect to the rotational motion. Note the span of the X axis of less than 1 m compared to the hundreds of meters in all other sequences. Moreover, for [Sequence 6 and 8](#), all estimations deviate in the beginning, and the accumulated error prevents a recovery of the accurate ground-truth position. Errors are carried from points with large black solid regions as in crossroads, and also shadows produced by buildings and trees

Table 1: Translation AAE, Rotation AAE and EPE ( $^{\circ}$ /frame)

	Yosemite			Fountain			Kitti 00-10	
	Trans. AAE	Rot. AAE	Rot. EPE	Trans. AAE	Rot. AAE	Rot. EPE	Trans. AAE	Rot. EPE
Raudies, 2014[1] using Brox[10]	2.4453	20.9280	0.0338	1.6380	1.4605	0.0102	14.7759	0.4300
Bruss, 1983[21] using Brox[10]	1.5285	28.5546	0.0126	1.1601	3.0578	<b>0.0028</b>	13.8722	0.3690
Kanatani, 1993[22] using Brox[10]	2.1252	13.8169	0.0182	1.4726	1.6544	0.0071	14.5035	0.4781
Raudies, 2014[1] using Sun[12]	2.2446	19.1784	0.0340	0.5120	0.5881	<b>0.0026</b>	14.0957	0.4504
Bruss, 1983[21] using Sun[12]	1.3245	26.7049	0.0125	1.0248	2.0480	0.0034	13.4256	0.3689
Kanatani, 1993[22] using Sun[12]	1.9742	10.8016	0.0122	<b>0.4012</b>	<b>0.5607</b>	0.0035	13.8342	0.4928
Raudies, 2014[1] using Vogel[11]	0.8085	21.5764	0.0109	0.7657	0.7893	0.0059	13.6363	0.4338
Bruss, 1983[21] using Vogel[11]	1.2289	26.4552	0.0122	1.5791	3.4248	0.0030	12.5186	0.1214
Kanatani, 1993[22] using Vogel[11]	0.7866	19.7755	0.0089	0.6864	1.0900	0.0041	13.4672	0.6690
Hui, 2013[30]	N/A	N/A	N/A	2.497	3.907	N/A	N/A	N/A
Yuan, 2013[32]	N/A	N/A	N/A	1.251	N/A	0.0699	N/A	N/A
Hui, 2015[31]	1.619	N/A	0.0282	2.371	N/A	0.0220	N/A	N/A
Yuan, 2015[33]	0.8803	N/A	0.0685	1.1866	N/A	0.050	N/A	N/A
Ji, 2006[29]	0.9589	5.4261	<b>0.0054</b>	1.4043	1.9327	<b>0.0022</b>	2.4923	0.0803
<b>Our approach: positive depth</b>	0.8436	<b>4.9055</b>	0.3799	1.2054	1.3528	0.2138	1.8225	<b>0.0613</b>
<b>Our approach: depth reconstructed</b>	<b>0.3640</b>	5.7749	0.3789	<b>0.4074</b>	<b>0.5615</b>	0.2139	<b>1.5340</b>	0.3545

(illumination changes severely impact the accuracy of image-based and normal flow based methods).

### 5.2. Comparison with methods that use optical flow and direct methods

Table 1 summarizes the comparison to several works in the literature using both normal flow and optical flow. The first rows show the error for three optical flow based methods: Raudies [2], Bruss [21], and Kanatani [22]. The optical flow is computed using three different methods: Brox [10], Sun [12], and Vogel [11].

The optical flow method of a) Sun [12], which ranked top 1 in 2014, is a variation of Horn & Schunck [9]. It uses a non-local smoothness regularization term based on median filtering, includes boundary and occlusion prediction to preserve motion, and uses an asymmetric hierarchical pyramid strategy to improve large motion estimations. b) The method of Vogel [11] is a variational method that uses Total Generalized Variation (TGV) regularization with a data term based on Census Transform with convex optimization (CSAD). c) The Brox [10] method targets specifically large displacements and is well suited for the Kitti dataset, where the inter-frame displacements reach up to 50 pixels.

The three selected optical-flow based methods for 3D motion estimation have been introduced in Section 1. These three methods showed the best per-

Table 2: Parameter set values

Vogel [11]			Sun [12]		
Parameter	Value	Description	Parameter	Value	Description
<i>cEps</i>	1.25/255	Threshold for Ternary Census	<i>max_iters</i>	3	Maximum number of iterations
<i>lambda</i>	12.333	Strength of data term	<i>gnc_iters</i>	2	Iterations for GNC (graduated non-convexity) formulation
<i>warps</i>	3	Number of warping steps	<i>texture</i>	true	Do texture decomposition
<i>pyramid_factor</i>	0.9	Scale in image pyramid	<i>median_filter_size</i>	5 x 5	Post-median filtering size
<i>innerIts</i>	10	Number of inner iterations	<i>area_hsz</i>	7	Half-window size for the weighted median filter
<i>ring</i>	2	Size of the patch for CENSUS	<i>sigma_i</i>	7	Half-window size for robust affine transformation
<i>dataTerm</i>	1	1 for CENSUS	<i>interpolation_method</i>	bi-cubic	Bicubic interpolation
			<i>doTV</i>	0	0 for total variation (TV)
			<i>stt</i>	0.5	Structure texture preprocessing
			<i>startResolution</i>	16	Minimum image size in pyramid
			<i>medFilt</i>	1	Post-warped median filter

formance and consistency in our exhaustive comparison in the lab. They have  
 380 been included to facilitate a comparison with direct approaches and seeking  
 completeness for future comparisons.

The parameters used in the optical flow estimation are given in Table 2.  
 No parameters are given for the optical flow method of Brox[10], since the  
 authors provide a library to execute their implementation. We use the original  
 385 values used in the authors' implementation for the parameters not listed in the  
 Table. The egomotion estimation methods also use the default parameters in  
 the implementations provided by the authors. The error values for the direct  
 methods are obtained from the cited publications. For the method of Sun [12],  
 preprocessing with a 5x5 Gaussian filter with  $\sigma = 1.5$  was used.

390 The next rows show methods that use normal flow for estimating egomotion.  
 Although, other methods exist in the literature ([17, 40, 52]), we only included  
 methods which have been evaluated on modern datasets. Hui *et al.*[30] com-  
 pute 3D motion from monocular normal flows and then estimate depth from  
 a stereo system without explicitly computing correspondences. They use two  
 395 constraints: the AFD, which is a relaxation of the constraint in Eq. (15) where a  
 voting mechanism punishes the estimates that result in negative depth (different  
 signs for the translational and rotational components). To make the constraint  
 more robust, several re-projections from one camera on the other are also con-  
 sidered. The second constraint, called AFM, affects the motion magnitude, and

400 is similar to the constraint in Eq. (16), that uses normal flows orthogonal to the translational component. In [31] the same authors propose a new solution based on the AFD constraint.

The method of Yuan *et al.*[32] first estimates the 3D motion using the constraint in Eq. (16), selecting the normal flow vectors orthogonal to the translational motion component. Then, they solve for the rotation with a more robust  
405 strategy, removing only solutions that do not lie in the half-plane consistent with the normal flow estimates. In Yuan *et al.*[33] authors present an updated version that relies on clustering for selecting the flows that satisfy the earlier described constraint and adding a step using RANSAC to refine the final estimations. We also have re-implemented the patch-based method for a single  
410 normal flow field described in [29]. This method, partitions the image into regular patches, and models the scene of each patch with a plane. We note that this planar constraint increases the number of parameters to be estimated.

Referring to Table 1, our approaches outperform all normal flow based methods for all cases (cases for which results are not reported in the literature are  
415 marked as N/A). Our method also outperforms all optical flow based methods, but the method of Kanatani [22] using the optical flow from Sun [12] on the the Fountain sequence. The results of our method may be further improved with an additional refinement step of the rotation after refining the translation, but  
420 at the cost of increased computation. We note the high accuracy of the method in [29], specifically for the rotational velocity. Let us emphasize the high performance of our method on the Kitti dataset, which is a real-world driving scenario in contrast to all other synthetic datasets previously used in the evaluation of direct 3D motion estimation methods.

425 Finally, let us point out that the normal flow is estimated using the spatial and temporal image derivatives. We set a threshold of 0.125 on the spatial gradient, with the images normalized between  $[0, 1]$ . This gradient was chosen empirically to maintain a sparsity below 10% of the full image resolution.

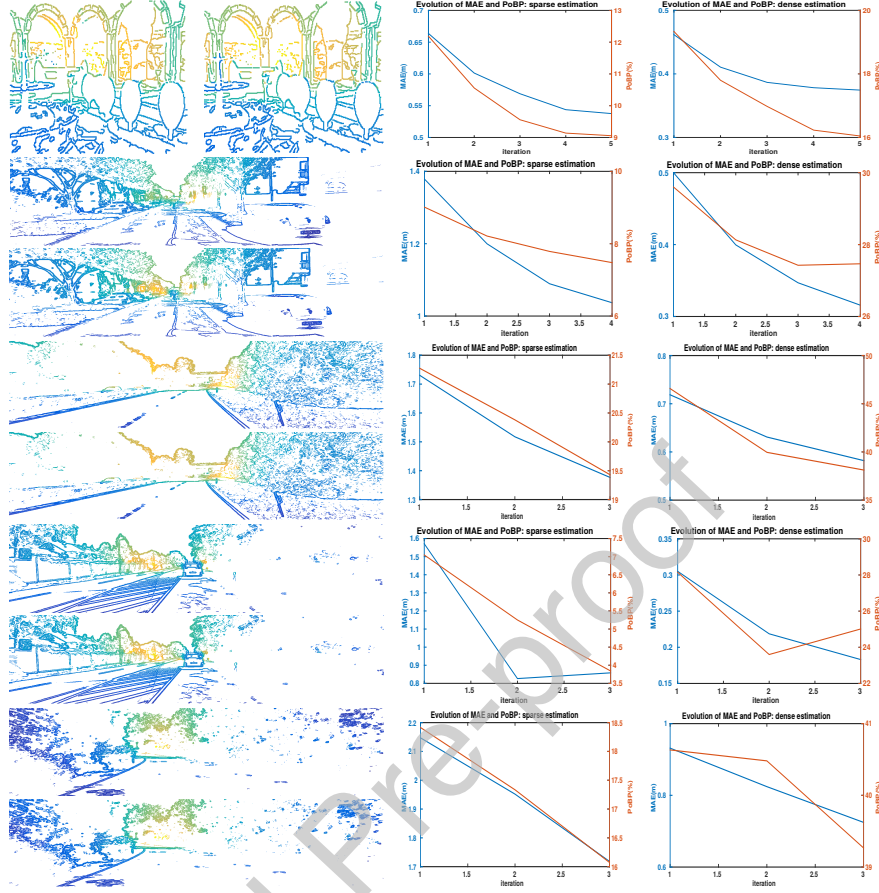


Figure 4: (Left) Sparse estimated 3D structure and ground-truth depth (dark blue represents closer objects and yellow the furthest ones); (right) evolution along iterative refinement of MAE (left in blue) and PoBP (right in red) for sparse and dense estimations. First row: frame 3 of Fountain sequence; remainder rows: (top) landscape image of estimated sparse depth and (bottom) sparse ground-truth for various example frames of the Kitti benchmark.

### 5.3. 3D reconstruction from motion

430 In this section we evaluate the accuracy of our method’s 3D geometry estimation by comparing to the disparity ground-truth (subject to availability). To obtain the scale of the depth, we use the provided ground-truth translational motion speed. Two metrics are used: 1) Mean Absolute Error (MAE), and 2) Percentage of Bad Points (PoBP).

435 Table 3 shows the accuracy of the 3D structure using these two error metrics, in addition to the density values which indicate the sparseness of the estimation

Table 3: 3D structure error metrics: MAE and PoBP

	MAE dense	PoBP dense	MAE sparse	PoBP sparse	density
Yosemite sequence [48]	N/A	N/A	N/A	N/A	10.54%
Fountain sequence [49]	0.359	15.60%	0.520	9.24% (1.44%)	9.60%
Kitti dataset [50]	0.800	32.91%	1.468	16.40% (1.80%)	10.95%

(common in sparse disparity estimation works such as [53, 54]). The first two columns provide the errors for the dense estimate after applying the inpainting method. The average error is below 0.4 m for the Fountain sequence (the maximum depth here is 7-8 m) and below 0.8 m for the Kitti dataset (with ranges up to 15-20 m). The third and fourth columns list the errors for our sparse estimate (about 10 % of the image resolution) before applying the regularization. Comparing the results in columns 1-4, we see that, because the inpainting helps recovering smooth surfaces where no gradients are found, it reduces the average absolute error. On the other hand, it negatively affects the PoBP, since the reconstruction process propagates the error at object contours due to the absence of values nearby. The values in parenthesis provide the PoBP for the whole image, not only the sparse estimate, which for both datasets is below 2%.

Fig. 4 illustrates the estimation of the 3D structure with examples from each sequence. The left and right image in row one, and top and bottom image in all other rows show the sparse estimated 3D structure and the sparse disparity ground-truth. The two plots on the right show the evolution of the MAE (in blue on the left axis) and the PoBP (in red on the right axis) along the refinement process. The number of iterations of this process depends on a convergence threshold that stops any further processing when there is not enough change between consecutive refined estimates. The first row shows frame 3 of the Fountain sequence, the second row shows frame 3 of the Yosemite sequence, and the last four rows show four frames from the Kitti dataset. As discussed in Table 3, the regularization process for the reconstruction negatively affects the PoBP, while it reduces the average MAE over all pixels. In most cases each refinement step reduces the MAE, for example 50% for sequence 5 of the Kitti dataset, while decreasing the PoBP.

The following values were used in the implementation. In the refinement of the egomotion using the 3D structure, the stopping criterium is: 1) a convergence threshold of 0.2 for the sum of differences between consecutive 3D structure estimates and 2) a maximum of 10 iterations. In the inpainting method, the only parameter is a multiplicative factor for the gradient of the curve that adjusts the smoothness; we set it to 0.01.

Regarding time performance, the bottleneck in our processing is the search for the translation. We used a 4-GHz Intel Core i7 computer with 8 GB of RAM. For the Fountain sequence, the average running time for the whole computation is 97.6 s, where 84.1 s correspond to the search, 13.05 s to the refinement (including three iterations until convergence), and the inpainting process requires 1.4 s. After parallelizing the search using 8 threads, the time for the search has been reduced to 13.04 s (and it is expected to be further reduced when using a massively parallel hardware such as a GPU). Comparing the time to optical flow based methods using the same computer, we found the following: Sun’s method [12] requires 61.56 s, Brox’s method [10] requires 5.95 s, and Vogel’s method [11] 29.14 s. After that, the time for running the 3D motion estimation methods is 0.76 s for Raudies’s [2] and 0.73 s for Kanatani’s [22]. However, as mentioned before, these methods rely on RANSAC for more refined estimations, and in this case Raudies’s requires 97.81 s and Kanatani’s about 200.81 s. Bearing in mind the accuracy of our direct method, the time performance is very similar to the conventional optical flow based methods.

## 6. Conclusions

We have introduced a new formulation of the depth positivity constraint. On the basis of this constraint, we proposed a direct method that allows for joint estimation of 3D motion and 3D structure using as input image gradients. The complete method consists of a non-linear optimization for the positive depth constraint using normal flow, followed by a refinement using a linear optimization on the depth. We showed that our method obtains higher accuracy in

motion estimation than other direct methods, and other optical flow based 3D motion estimation techniques. Furthermore, the estimated 3D geometry of the scene was shown of good quality.

495 Our results demonstrate that delaying the smoothness constraint, and estimating 3D motion globally in early stages of the structure-from-motion pipeline from minimal data, is a good choice. Although we used the new depth positivity constraint with normal flow, it also could be applied to optical flow. Thus our work may inspire further approaches that include the positivity constraint into 500 the 3D motion estimation. One possibility is to incorporate the constraint into a deep learning architecture for motion and structure estimation [55, 56].

### Acknowledgment

This work was supported by a *Juan de la Cierva* grant (IJCI-2014-21376), partially funded by the AEI Grant PID2019-109434RA-I00, the Research Network RED2018-102511-T, and the National Science Foundation under grants 505 SMA 1540917 and CNS 1544797.

### References

- [1] F. Raudies, H. Neumann, An efficient linear method for the estimation of ego-motion from optical flow, in: Joint Pattern Recognition Symposium, 510 Springer, 2009, pp. 11–20.
- [2] F. Raudies, H. Neumann, A review and evaluation of methods estimating ego-motion, Computer Vision and Image Understanding 116 (5) (2012) 606–633.
- [3] K. Pauwels, M. M. Van Hulle, Optimal instantaneous rigid motion estimation insensitive to local minima, Computer Vision and Image Understanding 515 104 (1) (2006) 77–86.
- [4] T. Zhang, C. Tomasi, On the consistency of instantaneous rigid motion estimation, International Journal of Computer Vision 46 (1) (2002) 51–79.

- [5] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, Monoslam: Real-time  
 520 single camera slam, *IEEE transactions on pattern analysis and machine  
 intelligence* 29 (6) (2007) 1052–1067.
- [6] R. Mur-Artal, J. D. Tardós, Orb-slam2: An open-source slam system for  
 monocular, stereo, and rgb-d cameras, *IEEE Transactions on Robotics*  
 33 (5) (2017) 1255–1262.
- [7] C. Forster, M. Pizzoli, D. Scaramuzza, Svo: Fast semi-direct monocular  
 525 visual odometry, in: *Robotics and Automation (ICRA)*, 2014 IEEE Inter-  
 national Conference on, IEEE, 2014, pp. 15–22.
- [8] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, *IEEE Transac-  
 tions on Pattern Analysis and Machine Intelligence*.
- [9] B. K. P. Horn, B. G. Schunck, Determining optical flow, *Artificial Intelli-  
 530 gence* 17 (1981) 185–203.
- [10] T. Brox, J. Malik, Large displacement optical flow: descriptor matching in  
 variational motion estimation, *IEEE transactions on pattern analysis and  
 machine intelligence* 33 (3) (2011) 500–513.
- [11] C. Vogel, S. Roth, K. Schindler, An evaluation of data costs for optical  
 535 flow, in: *German Conference on Pattern Recognition*, Springer, 2013, pp.  
 343–353.
- [12] D. Sun, S. Roth, M. Black, A quantitative analysis of current practices  
 in optical flow estimation and the principles behind them, *International  
 540 Journal of Computer Vision* 106 (2) (2014) 115–137.
- [13] B. K. Horn, E. Weldon, Direct methods for recovering motion, *International  
 Journal of Computer Vision* 2 (1) (1988) 51–76.
- [14] S. Negahdaripour, B. K. Horn, A direct method for locating the focus of  
 expansion, *Computer Vision, Graphics, and Image Processing* 46 (3) (1989)  
 545 303–326.

- [15] S. Carlsson, Sufficient image structure for 3-d motion and shape estimation, in: European Conference on Computer Vision, Springer, 1994, pp. 83–91.
- [16] C. Fermüller, Y. Aloimonos, The role of fixation in visual motion analysis, *International Journal of Computer Vision* 11 (2) (1993) 165–186.
- 550 [17] C. Fermüller, Passive navigation as a pattern recognition problem., *International Journal of Computer Vision* 14 (2) (1995) 147–158.
- [18] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge university press, 2003.
- [19] H. C. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, MA Fischler and O. Firschein, eds (1987) 61–62.
- 555 [20] B. Horn, Robot vision, MIT Press, 1986.
- [21] A. R. Bruss, B. K. Horn, Passive navigation, *Computer Vision, Graphics, and Image Processing* 21 (1) (1983) 3–20.
- 560 [22] K. Kanatani, 3-d interpretation of optical flow by renormalization, *International Journal of Computer Vision* 11 (3) (1993) 267–282.
- [23] J. Engel, T. Schöps, D. Cremers, Lsd-slam: Large-scale direct monocular slam, in: European Conference on Computer Vision, Springer, 2014, pp. 834–849.
- 565 [24] R. Ranftl, V. Vineet, Q. Chen, V. Koltun, Dense monocular depth estimation in complex dynamic scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4058–4066.
- [25] J. Aloimonos, C. M. Brown, Direct processing of curvilinear sensor motion from a sequence of perspective images, in: Proc. Workshop on Computer Vision: Representation and Control, Vol. 72, 1984, p. 77.
- 570

- [26] D. Sinclair, A. Blake, D. Murray, Robust estimation of egomotion from normal flow, *International Journal of Computer Vision* 13 (1) (1994) 57–69.
- [27] C. Fermüller, Y. Aloimonos, Qualitative egomotion, *International Journal of Computer Vision* 15 (1-2) (1995) 7–29.
- [28] T. Brodsky, C. Fermüller, Y. Aloimonos, Shape from video: Beyond the epipolar constraint, in: *Proceedings of the DARPA Image Understanding Workshop*, 1998.
- [29] H. Ji, C. Fermüller, A 3d shape constraint on video, *IEEE transactions on pattern analysis and machine intelligence* 28 (6) (2006) 1018–1023.
- [30] T.-W. Hui, R. Chung, Structure from motion directly from a sequence of binocular images without explicit correspondence establishment, in: *Image Processing (ICIP), 2013 20th IEEE International Conference on*, IEEE, 2013, pp. 3607–3611.
- [31] T.-W. Hui, R. Chung, Determining shape and motion from monocular camera: A direct approach using normal flows, *Pattern Recognition* 48 (2) (2015) 422–437.
- [32] D. Yuan, M. Liu, H. Zhang, Direct ego-motion estimation using normal flows, in: *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, IEEE, 2013, pp. 310–314.
- [33] D. Yuan, M. Liu, J. Yin, J. Hu, Camera motion estimation through monocular normal flow vectors, *Pattern Recognition Letters* 52 (2015) 59–64.
- [34] T.-W. Hui, K. N. Ngan, Dense depth map generation using sparse depth data from normal flow, in: *Image Processing (ICIP), 2014 IEEE International Conference on*, IEEE, 2014, pp. 3837–3841.
- [35] H. C. Longuet-Higgins, K. Prazdny, The interpretation of a moving retinal image, *Proceedings of the Royal Society of London B: Biological Sciences* 208 (1173) (1980) 385–397.

- [36] K. Daniilidis, H.-H. Nagel, Analytical results on error sensitivity of motion  
 600 estimation from two views, *Image and Vision Computing* 8 (4) (1990) 297–  
 303.
- [37] K. Kanatani, 3-d interpretation of optical flow by renormalization, *International Journal of Computer Vision* 11 (3) (1993) 267–282. doi:  
 10.1007/BF01469345.  
 605 URL <https://doi.org/10.1007/BF01469345>
- [38] R. Y. Tsai, T. S. Huang, Uniqueness and estimation of three-dimensional  
 motion parameters of rigid objects with curved surfaces, *Coordinated Science Laboratory Report no. UILU-ENG 81-2252, R-921*.
- [39] H. C. Longuet-Higgins, The visual ambiguity of a moving plane, *Proceedings of the Royal Society of London. Series B. Biological Sciences* 223 (1231)  
 610 (1984) 165–175.
- [40] T. Brodsky, C. Fermüller, Y. Aloimonos, Structure from motion: Beyond  
 the epipolar constraint, *International Journal of Computer Vision* 37 (3)  
 (2000) 231–258.
- [41] C. Fermüller, Y. Aloimonos, Observability of 3d motion, *International Journal of Computer Vision* 37 (1) (2000) 43–63.  
 615
- [42] C. Fermüller, Y. Aloimonos, Direct perception of three-dimensional motion  
 from patterns of visual motion, *Science* 270 (5244) (1995) 1973.
- [43] P. Corke, J. Lobo, J. Dias, An introduction to inertial and visual sensing,  
 620 *The International Journal of Robotics Research* 26 (6) (2007) 519–535.
- [44] R. H. Byrd, M. E. Hribar, J. Nocedal, An interior point algorithm for large-  
 scale nonlinear programming, *SIAM Journal on Optimization* 9 (4) (1999)  
 877–900.
- [45] R. A. Waltz, J. L. Morales, J. Nocedal, D. Orban, An interior algorithm  
 625 for nonlinear optimization that combines line search and trust region steps,  
*Mathematical programming* 107 (3) (2006) 391–408.

- [46] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, T. Darrell, A category-level 3d object dataset: Putting the kinect to work, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 141–165.
- [47] W. E. L. Grimson, From images to surfaces: A computational study of the human early visual system, MIT press, 1981.
- [48] J. Barron, Yosemite dataset, <http://www.csd.uwo.ca/faculty/barron/FTP/>, accessed: 2018-01-15 (2018).
- [49] F. Raudies, Fountain dataset, <http://cns.bu.edu/~fraudies/ScenesImages/ScenesImages.html>, accessed: 2018-01-15 (2018).
- [50] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
- [51] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, International journal of computer vision 47 (1-3) (2002) 7–42.
- [52] C. Silva, J. Santos-Victor, Robust egomotion estimation from the normal flow using search subspaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (9) (1997) 1026–1034.
- [53] J. Brandt, Improved accuracy in gradient-based optical flow estimation, International Journal of Computer Vision 25 (1) (1997) 5–22.
- [54] F. Barranco, M. Tomasi, J. Diaz, M. Vanegas, E. Ros, Parallel architecture for hierarchical optical flow estimation based on fpga, Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 20 (6) (2012) 1058–1067.
- [55] T. Zhou, M. Brown, N. Snavely, D. G. Lowe, Unsupervised learning of depth and ego-motion from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

- <sup>655</sup> [56] Z. Yin, J. Shi, Geonet: Unsupervised learning of dense depth, optical flow and camera pose, arXiv preprint arXiv:1803.02276.

**Francisco Barranco** received his MSc in Computer and Network Engineering in 2008 and his Ph.D. in Computer Science in 2012 from the University of Granada. He is a EU Marie Curie fellow at University of Maryland and works  
 660 for the Computer Science Department. His main research interests concern robotics, embedded real-time machine vision, bio-inspired processing, and cognitive vision. He has participated in different EU projects for adaptive learning and real-time computer vision.

**Cornelia Fermüller** is a Research Scientist at the University of Maryland  
 665 Institute for Advanced Computer Studies. She holds a Ph.D. from the Vienna University of Technology, Austria (1993) and an M.S. from the Graz University of Technology (1989), both in Applied Mathematics. Her research interest has been to understand principles of active vision systems and develop biological-inspired methods, especially in the area of motion. Her recent work is centered  
 670 on developing robot vision for interpreting human manipulation actions.

**Yiannis Aloimonos** (PhD 1987 Univ. of Rochester) is a Professor of Computational Vision and Intelligence at the Department of Computer Science of the University of Maryland at College Park and the Director of the Computer Vision Laboratory at the Institute for Advanced Computer Studies (UMIACS).  
 675 He received the Presidential Young Investigator Award (PECASE) for his work on Active Vision. He is interested in the integration of perception, action and cognition (bridging signals and symbols).

**Eduardo Ros** received his Ph.D. degree in 1997. He is currently Full Professor at the Department of Computer Architecture and Technology at the University of Granada. He leads an interdisciplinary lab, with interest in computational  
 680 neuroscience, neuromorphic engineering real-time processing, etc. He is also researching ultra-accurate timing systems in different industrial applications. In particular, his main research interests include ultra-accurate time transfer and synchronization of distributed instrumentation systems, simulation of spiking  
 685 neural networks, high performance computer processing, hardware implementation of digital circuits for real-time processing in embedded systems.