

# Optimal Multi-View Surface Normal Estimation Using Affine Correspondences

Dániel Baráth<sup>1</sup>, Ivan Eichhardt, and Levente Hajder

**Abstract**—An optimal, in the least squares sense, method is proposed to estimate surface normals in both stereo and multi-view cases. The proposed algorithm exploits exclusively photometric information via affine correspondences and estimates the normal for each correspondence independently. The normal is obtained as a root of a quartic polynomial. Therefore, the processing time is negligible. Eliminating the outliers, we propose a robust extension of the algorithm that combines maximum likelihood estimation and iteratively re-weighted least squares. The method has been validated on both synthetic and publicly available real-world datasets. It is superior to the state of the art in terms of accuracy and processing time. Besides, we demonstrate two possible applications: 1) using our algorithm as the seed-point generation step of patch-based multi-view stereo method, the obtained reconstruction is more accurate, and the error of the 3D points is reduced by 30% on average and 2) multi-plane fitting becomes more accurate applied to the resulting oriented point cloud.

**Index Terms**—Surface normal, affine features, multi-view.

## I. INTRODUCTION

**E**VEN though computer vision has been an intensively researched area for decades, several unsolved problems exist. The one, we aim at in this paper, is the analytic estimation of surface normals in a multi-view system exploiting solely photometric information, *i.e.* affine correspondences. The spatial relationship of the points is not considered thus achieving point-wise estimation without requiring dense clouds.

Several tasks, including surface reconstruction and segmentation, or object detection, require accurate surface normals. Benefiting from the higher-order information which they encode, surface reconstruction becomes more accurate and

robust. For example the widely-used Poisson-reconstruction technique [1], [2] is based on both the point coordinates and the normal. Having an oriented point cloud makes geometric primitive fitting, *e.g.* that of planes or cylinders, significantly easier due to the fact that less points are enough for the model-hypothesis generation. This number highly influences state-of-the-art multi-model fitting algorithms like PEARL [3] in terms of accuracy and processing time. As an example, plane fitting needs at least one oriented or three non-oriented points.

One of the first algorithms solving the surface normal estimation problem was the photometric stereo (PS) method [4]. Requiring totally controlled light conditions, the applicability of PS is limited into the laboratory. The original PS assumes Lambertian surface, thus not dealing with shiny materials, and estimates the normal using the so-called “Bidirectional Reflectance Distribution Function” [5] with known light-source parameters. However, several modifications, see [6], [7], have been proposed since then, making it more accurate and applicable to various materials.

Between two calibrated views, the normal estimation problem is often approached by decomposing the homographies of corresponding image patches [8], [9]. For calibrated views, a homography can be interpreted as the tangent plane of the surface at the observed 3D location. However, the decomposition itself is ambiguous as it was shown by several studies, *e.g.* in [10], and homography estimation cannot be done for each point correspondence independently. Thus, in general, these methods are applied to corresponding image regions supposing that the underlying surface patch is planar.

Köser [11] proposed a technique using local affine transformations. In brief, a local affinity is interpreted as the partial derivative, w.r.t. the image directions, of the underlying homography at the observed location. Therefore, it encodes higher-order geometric information, *i.e.* the surface normal. The method in [11] was the first which made the analytical point-wise normal estimation achievable between two views since local affinities can be measured by affine-covariant feature detectors, *e.g.* Hessian-Affine [12], Affine-SIFT [13] or MODS [14], for each correspondence separately. Benefiting from this approach, the ambiguity, to which the homography decomposition leads, disappeared. Barath *et al.* [15] proposed a method for optimal normal estimation. This paper extends their algorithm to the multi-view case.

Considering several views, an objective of several structure-from-motion (SfM) pipelines is to estimate the surface normals accurately since they contain fundamental information for

Manuscript received September 11, 2017; revised August 3, 2018 and December 12, 2018; accepted January 7, 2019. Date of publication January 25, 2019; date of current version May 14, 2019. The work of D. Barath was supported in part by the Hungarian Scientific Research Fund (OTKA KH\_126513 and K\_120499) and in part by the OP VVV Funded Project under Grant CZ.02.1.01/0.0/0.0/16\_019/0000765. The work of I. Eichhardt was supported by the Hungarian Scientific Research Fund under Grant OTKA 120499. The work of L. Hajder was supported in part by the Hungarian Government and in part by the European Social Fund under Grant EFOP-3.6.3-VEKOP-16-2017-00001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Abd-Krim K. Seghouane. (*Corresponding author: Dániel Baráth.*)

D. Baráth is with the Centre for Machine Perception, Czech Technical University in Prague, Prague 160 00, Czech Republic, and also with the Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary (e-mail: barath.daniel@sztaki.mta.hu).

I. Eichhardt is with the Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary.

L. Hajder is with the Department of Algorithms and Their Applications, Eötvös Loránd University, Budapest 1053, Hungary.

Digital Object Identifier 10.1109/TIP.2019.2895542

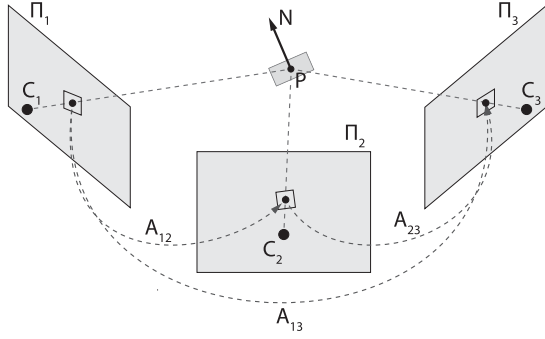


Fig. 1. Three cameras observing a point  $P$  on a plane with normal  $N$ . The neighboring pixels of the projected points between the  $i$ th and  $j$ th views are related by a local affine transformation  $A_{ij}$ .

the latter surface reconstruction. The well-known algorithm, called Patch-based Multi-View Stereo (PMVS) proposed by Furukawa and Ponce [16], [17], solves the problem as an optimization numerically refining the plane parameters for minimizing a joint photometric cost function. The cost is based on zero-mean cross-correlation applied to patches, each transformed by the homography which the plane induces. Reference [18] approaches the problem similarly to PMVS, assuming that the surfaces can be represented by local planar patches. It proposes a unified cost function considering both geometric and photometric terms. These methods obtain accurate surface normals, however, they are sensitive to the size of the patch for which the photometric cost is computed. Being solved numerically they are relatively slow. Also, due to having no proofs of the convexity of the minimized costs, they do not guarantee global optimum and the results depend on an initial parameter setting.

The contributions of the paper are: (i) we propose an analytic multi-view normal estimation technique which is optimal in the least squares sense and uses local affine transformations (see Fig. 1). First, we show the relationship of affinities and surface normals considering two views, then this approach is extended. This is the first analytic solution applicable to the multi-view case. The equations are not linearized, thus, the globally optimal solution is carried out efficiently as a root of a fourth-order polynomial thus achieving fast calculation. (ii) Reflecting the fact that the estimation of local affinities is sensitive to the view angle, thus a measured set of affinities might contain outliers, we propose a robust estimation technique. It is reported both on synthesized and real world tests, that the proposed method outperforms the state-of-the-art in terms of accuracy and processing time. (iii) Besides, we demonstrate the applicability of the method on two problems: replacing the seed-point generation step of PMVS with the proposed approach leads to more accurate reconstruction; and multi-plane fitting becomes more robust applied to the resulting oriented point cloud.

## II. THEORETICAL BACKGROUND

In this section, we discuss the relationship of local affine transformations and surface normals [15]. Assume that a surface point  $[x \ y \ z]^T$  is observed by two cameras. The camera model is arbitrary. The projected image points  $\mathbf{p}_1 = [u_1 \ v_1]^T$

and  $\mathbf{p}_2 = [u_2 \ v_2]^T$  are calculated using the  $3D \rightarrow 2D$  projection function  $\Pi_i$  as  $[u_i \ v_i]^T = \Pi_i(x, y, z)$ , where  $i \in \{1, 2\}$  denotes the view. Affine transformation  $\mathbf{A}$ , mapping the infinitesimally close neighborhood of  $\mathbf{p}_1$  to that of  $\mathbf{p}_2$ , is defined by the Jacobian of the surface projections through  $\Pi_1$  and  $\Pi_2$  as follows:

$$\mathbf{A} = \mathbf{J}_2 \mathbf{J}_1^{-1} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}, \quad (1)$$

if the surface is written by parametric representation. As it is written in Appendix, the affine transformation  $\mathbf{A}$  between two views can be determined if two projective functions  $\Pi^i$ ,  $i \in \{1, 2\}$  and surface normal  $\mathbf{n}$  are given:

$$\mathbf{A} = \frac{1}{(\Pi_x^1)^T[\mathbf{n}] \times \Pi_y^1} \begin{bmatrix} (\Pi_x^2)^T[\mathbf{n}] \times \Pi_y^1 & (\Pi_x^1)^T[\mathbf{n}] \times \Pi_x^2 \\ (\Pi_y^2)^T[\mathbf{n}] \times \Pi_y^1 & (\Pi_x^1)^T[\mathbf{n}] \times \Pi_y^2 \end{bmatrix}. \quad (2)$$

The upper index denotes the view, the lower the spatial coordinates:  $x$ ,  $y$ , or  $z$ . As it is discussed in detail in the appendix, by building the Jacobians using gradient vectors  $\nabla \Pi_u^i$  and  $\nabla \Pi_v^i$ , denoting image coordinates by  $u$  and  $v$ , and multiplying them, local affine transformation  $\mathbf{A}$  becomes

$$\mathbf{A} = \mathbf{J}_2 \mathbf{J}_1^{-1} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} = \frac{1}{\mathbf{n}^T \mathbf{w}_5} \begin{bmatrix} \mathbf{n}^T \mathbf{w}_1 & \mathbf{n}^T \mathbf{w}_2 \\ \mathbf{n}^T \mathbf{w}_3 & \mathbf{n}^T \mathbf{w}_4 \end{bmatrix}, \quad (3)$$

where

$$\begin{aligned} \mathbf{w}_1 &= \nabla \Pi_v^1 \times \nabla \Pi_u^2, & \mathbf{w}_2 &= \nabla \Pi_u^2 \times \nabla \Pi_u^1, \\ \mathbf{w}_3 &= \nabla \Pi_v^1 \times \nabla \Pi_v^2, & \mathbf{w}_4 &= \nabla \Pi_v^2 \times \nabla \Pi_u^1, \\ \mathbf{w}_5 &= \nabla \Pi_v^1 \times \nabla \Pi_v^1. \end{aligned} \quad (4)$$

For the perspective camera of the  $i$ -th frame, the projection is written as  $[u_i \ v_i \ 1]^T = (1/s_i) \mathbf{P}_i [x \ y \ z \ 1]^T$ , where  $\mathbf{P}_i$  ( $i \in \{1, 2\}$ ) is the projection matrix,  $s_i = p_{i,31}x + p_{i,32}y + p_{i,33}z + p_{i,34}$  is the projective depth,  $u_i$  and  $v_i$  are the projected coordinates in the  $i$ th image, and  $[x \ y \ z \ 1]^T$  is the homogeneous 3D point. The gradients of the projection formulas w.r.t. to the spatial directions are as follows:

$$\begin{aligned} \frac{\partial u_i}{\partial x} &= \frac{1}{s_i} (p_{i,11} - u_i p_{i,31}), & \frac{\partial u_i}{\partial y} &= \frac{1}{s_i} (p_{i,12} - u_i p_{i,32}), \\ \frac{\partial u_i}{\partial z} &= \frac{1}{s_i} (p_{i,13} - u_i p_{i,33}), & \frac{\partial v_i}{\partial x} &= \frac{1}{s_i} (p_{i,21} - v_i p_{i,31}), \\ \frac{\partial v_i}{\partial y} &= \frac{1}{s_i} (p_{i,22} - v_i p_{i,32}), & \frac{\partial v_i}{\partial z} &= \frac{1}{s_i} (p_{i,23} - v_i p_{i,33}). \end{aligned}$$

Therefore, the gradient vectors are written as

$$\nabla \Pi_{i,u} = \frac{1}{s_i} \begin{bmatrix} p_{i,11} - u_i p_{i,31} \\ p_{i,12} - u_i p_{i,32} \\ p_{i,13} - u_i p_{i,33} \end{bmatrix}, \quad \nabla \Pi_{i,v} = \frac{1}{s_i} \begin{bmatrix} p_{i,21} - v_i p_{i,31} \\ p_{i,22} - v_i p_{i,32} \\ p_{i,23} - v_i p_{i,33} \end{bmatrix}.$$

Eq. 3 determines the relationship of surface normals and local affine transformations for the perspective camera model. We will use this relationship to define the optimal solvers for both the two-view and multi-view cases.

Note that if the projective depth  $s_i$  is unknown, but the upper left  $3 \times 3$  submatrices of the projection matrices  $\mathbf{P}_1$  and  $\mathbf{P}_2$  are known, the gradient vectors can be calculated up to an unknown scale – this scale is the multiplicative inverse of the projective depth  $s_i$ . Also note that vectors  $\mathbf{w}_1, \dots, \mathbf{w}_4$  are

scaled by  $s_1$   $s_2$  while  $\mathbf{w}_5$  by  $s_1$   $s_1$ . Therefore, the surface normal is independent of the translation between the two cameras since the last columns of the projection matrices are the product of the intrinsic parameters and the translation.

### III. MULTI-VIEW OPTIMAL SURFACE NORMALS

Optimal solvers are proposed for the stereo and multi-view cases in this section. Then we propose a robust extension of the algorithm minimizing the effect of the outliers.

#### A. Stereo Case

In this section, we show that a surface normal  $\mathbf{n} = [n_x \ n_y \ n_z]^T$  can be estimated optimally, in the least squares sense, exploiting a local affinity. The solution for the two-view case was first presented in [15]. Suppose that an affine correspondence  $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{A})$  obtained by *e.g.* an affine-covariant feature detector is given in two images. The optimization problem is written by reformulating Eq. 3 as follows:

$$\arg \min_{\mathbf{n}} \sum_{k=1}^4 \left( \frac{\mathbf{n}^T \mathbf{w}_k}{\mathbf{n}^T \mathbf{w}_5} - a_k \right)^2, \quad (5)$$

where the only unknowns are the coordinates of  $\mathbf{n}$ . Note that the four equations can be linearized multiplying each by  $\mathbf{n}^T \mathbf{w}_5$ , however, the linearization distorts the original signal-noise ratio leading to noise-sensitive estimates. Also note that the minimization of the Frobenius-norm has not merely an algebraic but a geometric interpretation as well in case of affine transformations. It is discussed by [19] in depth.

Such kind of optimization problems are usually solved by Lagrange-multipliers, however, in the current case the derivatives would be difficult to solve. Therefore, we exploit another linear solution. As the basic equations are valid if non unit-length surface normals are applied, other constraints can be introduced. Due to simplicity, we fix the sum of the coordinates to be unit, *i.e.*  $n_x + n_y + n_z = 1$ . Then the normal can be written as  $\mathbf{n} = [n_x \ n_y \ 1 - n_x - n_y]^T$ . Applying this constraint to Eq. 5, we get the following equation

$$\arg \min_{\mathbf{m}} \sum_{k=1}^4 \left( \frac{\mathbf{m}^T \mathbf{q}_k + w_{k,z}}{\mathbf{m}^T \mathbf{q}_5 + w_{5,z}} - a_k \right)^2, \quad (6)$$

where  $\mathbf{m} = [n_x \ n_y]^T$ ,  $\mathbf{q}_i = [w_{i,x} - w_{i,z} \ w_{i,y} - w_{i,z}]^T$ , and  $w_{i,x}$ ,  $w_{i,y}$ ,  $w_{i,z}$  are the  $x$ ,  $y$ ,  $z$  coordinates of  $\mathbf{w}_i$ .

If  $n_x + n_y + n_z \approx 0$ , constraint  $n_x + n_y + n_z = 1$  is not applicable. The length of the obtained normal is near infinity and, thus, this degeneracy can be detected. Even though this case cannot be handled by the original method, the constraint is replaced by one of the following ones:  $n_x = 1$ ,  $n_y = 1$  or  $n_z = 1$ . These cases are solved independently and the solution with the minimum cost is chosen. The modification of the cost function, defined in Eq. 6, is straightforward, only the corresponding coordinates are replaced.

The optimal solution, in the least squares sense, is where the derivative w.r.t.  $\mathbf{m}$  is equal to zero:

$$\sum_{k=1}^4 \beta_k \mathbf{r}_k = 0,$$

where  $\beta_k = (\mathbf{m}^T \mathbf{q}_k + w_{k,z}) / (\mathbf{m}^T \mathbf{q}_5 + w_{5,z}) - a_k$ ,  $\mathbf{r}_k = ((\mathbf{m}^T \mathbf{q}_5 + w_{5,z}) \mathbf{q}_k - (\mathbf{m}^T \mathbf{q}_k + w_{k,z}) \mathbf{q}_5) / ((\mathbf{m}^T \mathbf{q}_5 + w_{5,z})^2)$ . Note that  $\mathbf{r}_k$  is a two-dimensional vector consisting of the expressions regarding to both coordinates of vector  $\mathbf{m}$ . After elementary modifications, including the multiplication by the denominator, the following formula is obtained:

$$\sum_{k=1}^4 s_k \begin{bmatrix} \mathbf{m}^T (\mathbf{q}_5 q_{k,x} - \mathbf{q}_i q_{5,x}) + w_{5,z} q_{k,x} - w_{k,z} q_{5,x} \\ \mathbf{m}^T (\mathbf{q}_5 q_{k,y} - \mathbf{q}_i q_{5,y}) + w_{5,z} q_{k,y} - w_{k,z} q_{5,y} \end{bmatrix} = 0,$$

where  $\mathbf{s} = \mathbf{m}^T (\mathbf{q}_k - \mathbf{q}_k \mathbf{q}_5) + w_{k,z} - a_k q_{5,z}$ . Replacing  $\mathbf{m}$  with its coordinates, the equation becomes

$$\sum_{k=1}^4 (\Omega_k n_x + \Psi_k n_y + \Gamma_k) \begin{bmatrix} \Omega_k^1 n_x + \Psi_k^1 n_y + \Gamma_k^1 \\ \Omega_k^2 n_x + \Psi_k^2 n_y + \Gamma_k^2 \end{bmatrix} = 0,$$

where

$$\begin{aligned} \Omega_k &= q_{k,x} - q_{5,x} a_k, & \Psi_k &= q_{k,y} - q_{5,y} a_k, \\ \Gamma_k &= w_{k,z} - a_k w_{5,z}, & \Omega_{k,1} &= 0, \\ \Psi_{k,1} &= q_{5,y} q_{k,x} - q_{k,y} q_{5,x}, & \Gamma_{k,1} &= w_{5,z} q_{k,z} - w_{k,z} q_{5,x}, \\ \Omega_{k,2} &= q_{5,x} q_{k,y} - q_{k,x} q_{5,y}, & \Psi_{k,2} &= 0, \\ \Gamma_{k,2} &= w_{5,z} q_{k,y} - w_{k,z} q_{5,y}. \end{aligned}$$

The rows of the vector equation yield two quadratic equation written in implicit form as

$$\sum_{k=1}^4 A_{k,1} n_x^2 + B_{k,1} n_y^2 + C_{k,1} n_x n_y + D_{k,1} n_x + E_{k,1} n_y + F_{k,1} = 0,$$

$$\sum_{k=1}^4 A_{k,2} n_x^2 + B_{k,2} n_y^2 + C_{k,2} n_x n_y + D_{k,2} n_x + E_{k,2} n_y + F_{k,2} = 0,$$

where

$$\begin{aligned} A_{k,l} &= \Omega_k \Omega_{k,l}, & B_{k,l} &= \Psi_k \Psi_{k,l}, \\ C_{k,l} &= \Omega_{k,l} \Psi_k + \Psi_{k,l} \Omega_k, & D_{k,l} &= \Omega_{k,l} \Gamma_k + \Gamma_{k,l} \Omega_k, \\ E_{k,l} &= \Psi_{k,l} \Gamma_k + \Gamma_{k,l} \Omega_k, & F_{k,l} &= \Gamma_k \Gamma_{k,l}, \end{aligned}$$

Due to  $\Omega_{k,1} = \Psi_{k,2} = 0$ ,  $A_{k,1} = B_{k,2} = 0$  ( $l \in \{1, 2\}$ ).

The summation can be eliminated from the equation by adding up the coefficients separately, *e.g.*  $\hat{B}_1 = \sum_{k=1}^4 B_{k,1}$ . Thus the resulting equations are as follows:

$$\hat{B}_1 n_y^2 + \hat{C}_1 n_x n_y + \hat{D}_1 n_x + \hat{E}_1 n_y + \hat{F}_1 = 0, \quad (7)$$

$$\hat{A}_2 n_x^2 + \hat{C}_2 n_x n_y + \hat{D}_2 n_x + \hat{E}_2 n_y + \hat{F}_2 = 0. \quad (8)$$

This polynomial system is straightforward to solve, thus applying a sophisticated polynomial solver, *e.g.* Groebner-basis [20], is unnecessary. We express parameter  $n_y$  from Eq. 8 as

$$n_y = -\frac{\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2}{\hat{C}_2 n_x + \hat{E}_2}. \quad (9)$$

Substituting the expression of  $n_y$  from Eq. 9 into Eq. 7 and multiplying by the denominator leads to

$$\begin{aligned} & \hat{B}_1 (\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2)^2 + \hat{F}_1 (\hat{C}_2 n_x + \hat{E}_2)^2 \\ & - \hat{C}_1 (\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2) (\hat{C}_2 n_x + \hat{E}_2) + \hat{D}_1 x (\hat{C}_2 n_x + \hat{E}_2)^2 \\ & - \hat{E}_1 (\hat{A}_2 n_x^2 + \hat{D}_2 n_x + \hat{F}_2) (\hat{C}_2 n_x + \hat{E}_2) = 0. \end{aligned}$$

The coefficients of all the monomials ( $n_x^4$ ,  $n_x^3$ ,  $n_x^2$ ,  $n_x^1$ , and  $n_x^0$ ) in the above equation are as follows.

$$\begin{aligned} n_x^4 &: \hat{B}_1 \hat{A}_2^2 - \hat{C}_1 \hat{A}_2 \hat{C}_2, \\ n_x^3 &: 2\hat{B}_1 \hat{A}_2 \hat{D}_2 - \hat{C}_1 \hat{A}_2 \hat{E}_2 - \hat{C}_1 \hat{D}_2 \hat{C}_2 + \hat{D}_1 \hat{C}_2^2 - \hat{E}_1 \hat{A}_2 \hat{C}_2, \\ n_x^2 &: \hat{B}_1 \hat{D}_2^2 + 2\hat{B}_1 \hat{A}_2 \hat{F}_2 - \hat{C}_1 \hat{D}_2 \hat{E}_2 - \hat{C}_1 \hat{F}_2 \hat{C}_2 \\ &\quad + 2\hat{D}_1 \hat{C}_2 \hat{E}_2 - \hat{E}_1 \hat{A}_2 \hat{E}_2 - \hat{E}_1 \hat{D}_2 \hat{C}_2 + \hat{F}_1 \hat{C}_2^2, \\ n_x^1 &: 2\hat{B}_1 \hat{D}_2 \hat{F}_2 - \hat{C}_1 \hat{F}_2 \hat{E}_2 + \hat{D}_1 \hat{E}_2^2 - \hat{E}_1 \hat{D}_2 \hat{E}_2 \\ &\quad - \hat{E}_1 \hat{F}_2 \hat{C}_2 + 2\hat{F}_1 \hat{C}_2 \hat{E}_2, \\ n_x^0 &: \hat{B}_1 \hat{F}_2^2 - \hat{E}_1 \hat{F}_2 \hat{E}_2 + \hat{F}_1 \hat{E}_2^2. \end{aligned}$$

This fourth-order polynomial equation can be solved by any polynomial solver toolbox, *e.g.* Matlab *roots* or OpenCV *solvePoly* methods. Coordinate  $n_y$  is then obtained using Eq. 9 and finally,  $n_z = 1 - n_x - n_y$ . To select the best out of the candidate real roots, we choose the one minimizing Eq. 5.

Summarizing this section, the coordinates of the surface normal can be optimally estimated in closed-form from two views as the roots of a fourth-order polynomial. Our method does not require linearizing any equations.

### B. Multi-View Case

Given a sequence of points in  $M \geq 2$  images with local affinities between every pair – this is a realistic assumption since affine-covariant feature detectors estimate Jacobian  $\mathbf{J}$  for each image independently, thus affinity  $\mathbf{A}_{ij}$  mapping from the  $i$ th to  $j$ th images is calculated as  $\mathbf{J}_j \mathbf{J}_i^{-1}$ . Extending Eq. 5 to more image pairs, the optimization problem becomes

$$\arg \min_{\mathbf{n}} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \sum_{k=1}^4 \left( \frac{\mathbf{n}^T \mathbf{w}_{ij,k}}{\mathbf{n}^T \mathbf{w}_{ij,5}} - a_{ij,k} \right)^2, \quad (10)$$

where each vector  $\mathbf{w}_{ij}$  is calculated similarly to Eq. 4 using the coordinates in the  $i$ th and  $j$ th images. It can be seen that the inner summation of the coefficients leads to two quadratic equations (Eqs. 7, 8), and the outer two is basically the summation of these equations over the possible view pairs:

$$\begin{aligned} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \hat{B}_{ij,1} n_y^2 + \hat{C}_{ij,1} n_x n_y \\ + \hat{D}_{ij,1} n_x + \hat{E}_{ij,1} n_y + \hat{F}_{ij,1} = 0, \end{aligned} \quad (11)$$

$$\begin{aligned} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \hat{A}_{ij,2} n_x^2 + \hat{C}_{ij,2} n_x n_y \\ + \hat{D}_{ij,2} n_x + \hat{E}_{ij,2} n_y + \hat{F}_{ij,2} = 0. \end{aligned} \quad (12)$$

These two equations can be formulated as

$$\hat{B}_1 y^2 + \hat{C}_1 n_x n_y + \hat{D}_1 n_x + \hat{E}_1 n_y + \hat{F}_1 = 0, \quad (13)$$

$$\hat{B}_2 y^2 + \hat{C}_2 n_x n_y + \hat{D}_2 n_x + \hat{E}_2 n_y + \hat{F}_2 = 0, \quad (14)$$

where

$$\hat{S}_k = \sum_{i=1}^{N-1} \sum_{j=i+1}^N \hat{S}_{ij,k} \quad k \in \{1, 2\}, \quad S \in \{B, C, D, E, F\}.$$

Thus the solution is given as the intersection of the summed equations (Eqs. 13, 14) in a fairly similar manner to that of the

two-view case. Note that the normalization of the coefficients is necessary to avoid numerical instability. Another note that the missing data problem, *i.e.* when information is not given for every image pair, can be resolved by introducing weight  $q_{ij}$  into Eq. 10. Weight  $q_{ij}$  is zero if there is no correspondence between the  $i$ th and  $j$ th views and one otherwise.

### C. Robust Estimation

Reflecting the fact that the local affinities might be contaminated by noise and contain outliers, we propose a robust estimation process in this section. The proposed algorithm consists of three major steps: **(a)** inconsistency check, **(b)** outlier filtering and **(c)** iteratively re-weighted least squares. In the rest of this section, we consider the estimation of only one surface normal thus having a single point tracked along an image sequence. The estimation remains point-wise, therefore, the spatial relationships in the reconstructed oriented point cloud is not exploited here.

1) *Inconsistency Check*: The aim of this step is to remove all view pairs for which the indicated surface normals do not satisfy geometric requirements. The proposed constraint is based on the assumption that the observed 3D point lies on a continuous surface which cannot be observed from behind. Therefore, for each camera, the surface normal must point towards a point of a hemisphere, having unit radius, around the 3D point which is the closest to the camera center (see the left plot of Fig. 3). For  $N$  views, the normal must be in the intersection of  $N$  hemispheres.

To remove outlier view pairs, we first estimate normal  $\mathbf{n}_{ij}$  for all  $i$ th and  $j$ th matched views ( $i, j \in [1, N], i \neq j$ ), *i.e.* the possible 2-combinations of the view matches. Then normal

$$\mathbf{n}_{ij} \text{ is outlier} = \exists \text{ view}_k, \text{ view}_m; k \neq m; k, m \in [1, N] : \langle \mathbf{n}_{ij}, \mathbf{v}_k \rangle \cdot \langle \mathbf{n}_{ij}, \mathbf{v}_m \rangle < 0.$$

As a result of this verification step, a set of view pairs are labeled as outliers and omitted from the latter steps. Note that this step considers a similar constraint to that of back-face culling which is widely-used in computer graphics.

2) *Outlier Filtering*: To select a set of inlier correspondences supporting the most likely normal, we applied MLESAC [21]. In each iteration, it takes a minimal sample, an affine correspondence, and estimates a surface normal. Then it selects the inlier set maximizing the likelihood of the estimated normal. As the error function, we used the squared Frobenius-norm of the matrix difference of the estimated and measured affine transformations. In the further steps we do not consider surface normals having less than two inliers.

3) *Iteratively Re-Weighted Least Squares*: The last part obtaining the final surface normal is an iteratively re-weighted least squares algorithm [22] (visualized in Fig. 2) applied to the inlier set provided by the previous steps. First, all weights are set to 1.0 and the indicated normal is computed applying the multi-view algorithm. Then, in each step of the alternation, the weights for the view pairs are re-calculated on the basis of the error of the estimated normal. Each weight  $q_{ij}$  regarding the  $i$ th and  $j$ th views affects the indicated quadratic equations



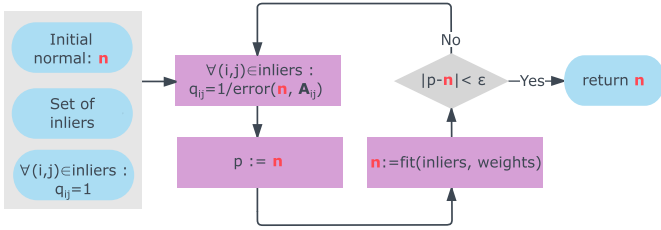


Fig. 2. Iteratively re-weighted least squares applied to the view pairs provided by steps (a) and (b) of the robust estimation. Weights  $q_{ij}$  are initialized to 1. The normal estimation is iterated using the current weights and the weighted equations (Eqs. 15, 16) until convergence.

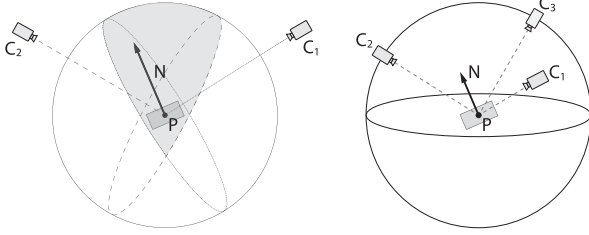


Fig. 3. **(Left)** The proposed geometric constraint demonstrated by two views. A hemisphere is selected by each camera (denoted by different dashed lines) around the observed point. The surface normal must be in the intersection of these hemispheres. **(Right)** The setup for the synthesized tests. The cameras are put in a random point of a sphere.

(the inner part of Eqs. 11, 12) by multiplying them as follows:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^N q_{ij} (\hat{B}_{ij,1} n_x^2 + \hat{C}_{ij,1} n_x n_y + \hat{D}_{ij,1} n_x + \hat{E}_{ij,1} n_y + \hat{F}_{ij,1}) = 0, \quad (15)$$

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^N q_{ij} (\hat{A}_{ij,2} n_x^2 + \hat{C}_{ij,2} n_x n_y + \hat{D}_{ij,2} n_x + \hat{E}_{ij,2} n_y + \hat{F}_{ij,2}) = 0. \quad (16)$$

After testing most of the state-of-the-art robust estimation techniques, we found that these three steps are the the most accurate for this problem.

#### IV. EXPERIMENTAL RESULTS

In this section, the performance of the proposed method is evaluated both on synthesized and real world tests.

##### A. Synthesized Tests

In order to test the proposed method in a fully controlled environment,  $N$  cameras were generated by their projection matrices looking towards the origin, each located in a random surface point of a 5-radius sphere. Then a random 3D oriented point, at most one unit far from the origin and with random normal, was projected onto the cameras. See the right plot of Fig. 3. The local affine transformation was calculated from the ground truth surface normal using Eq. 3. Finally, zero-mean Gaussian noise with  $\sigma$  standard deviation was added to both the point locations and affine parameters. The reported results are computed as the mean of 500 runs for each test case.

The competitor algorithms are the two-view optimal method proposed in this paper, the techniques of Köser [11] and Barath *et al.* [15]. Since they are 2-view methods and, thus, cannot be applied directly to multiple views, we applied them to every possible view pair. The final results are calculated as the means, in the spherical domain, of the obtained normals. Reflecting the fact that the normals are insensitive to the scale, *e.g.* to multiplier  $-1$ , they were normalized and it was made sure that they look towards the same direction.

Figs. 4(a), 4(b), 4(c) and 4(d) plot the angular error (in degrees) as a function of the noise  $\sigma$  for 3, 5, 10 and 25 views, respectively. It can be seen that the proposed method outperforms the competitor algorithms.

Fig. 4(e) shows the angular error as a function of the view number with fixed  $\sigma = 0.5$  pixel noise. It can be seen that the proposed method is consistent - the more samples are given, the lower error is achieved -, and converges to the ground truth normal faster than the other methods.

Figs. 4(g), 4(h) and 4(i) compare the robust version of the proposed algorithm to the original one with  $\sigma$  set to 0.1, 0.5 and 1.0 pixels, respectively. For these tests  $I \in [2, 15]$  views were generated, and  $15 - I$  outliers (random point correspondences and affinities) were added. For example, if  $I = 10$ , *i.e.* 10 inlier and 5 outlier views are given, the outlier percentage is calculated as  $1 - \binom{10}{2} / \binom{15}{2} \approx 0.57$ . In the figures, the horizontal axis reports the outlier ratio and the vertical one shows the mean angular error of the results. It can be seen that the robust version of the proposed algorithm is able to fully overcome at most 50 – 60% outlier ratio, and significantly reduces the error even for higher noise level.

The mean processing times of the methods are reported in Fig. 4(f) plotted as the function of the view number. Due to the pair-wise parameter calculation, the time demands of all methods show a quadratic trend, however, the proposed one is significantly faster for more views than the competitors, *e.g.* processing 25 views lasts  $\approx 0.03$  seconds in Matlab.

##### B. Real World Tests

To test the proposed method on real world data we used the publicly available benchmarking datasets of Strecha *et al.* [23], Pusztai and Hajder [24] and ETH3D [25]. The dataset<sup>1</sup> of [23] consists of several images of size  $3072 \times 2048$  of buildings. Both the intrinsic and extrinsic parameters are given for all images, the dense point cloud for each scene is obtained using a LiDAR sensor. The images of [24] are captured by a turn-table equipment, the cameras are calibrated and the ground truth point clouds are estimated using a structured light scanner. ETH3D<sup>2</sup> contains image sequences captured by both HD and mobile cameras and 3D point clouds obtained by laser scanner. For all datasets, the ground truth surface normals are estimated using the dense point clouds, fitting a paraboloid to the neighborhood of each point.

Obtaining affine correspondences, the Tree-Based Morse Regions [26] (TBMR) algorithm was applied to extract the

<sup>1</sup>Available at <http://cvlabwww.epfl.ch/data/multiview/denseMVS.html>

<sup>2</sup>Available at <https://www.eth3d.net/datasets#high-res-multi-view>

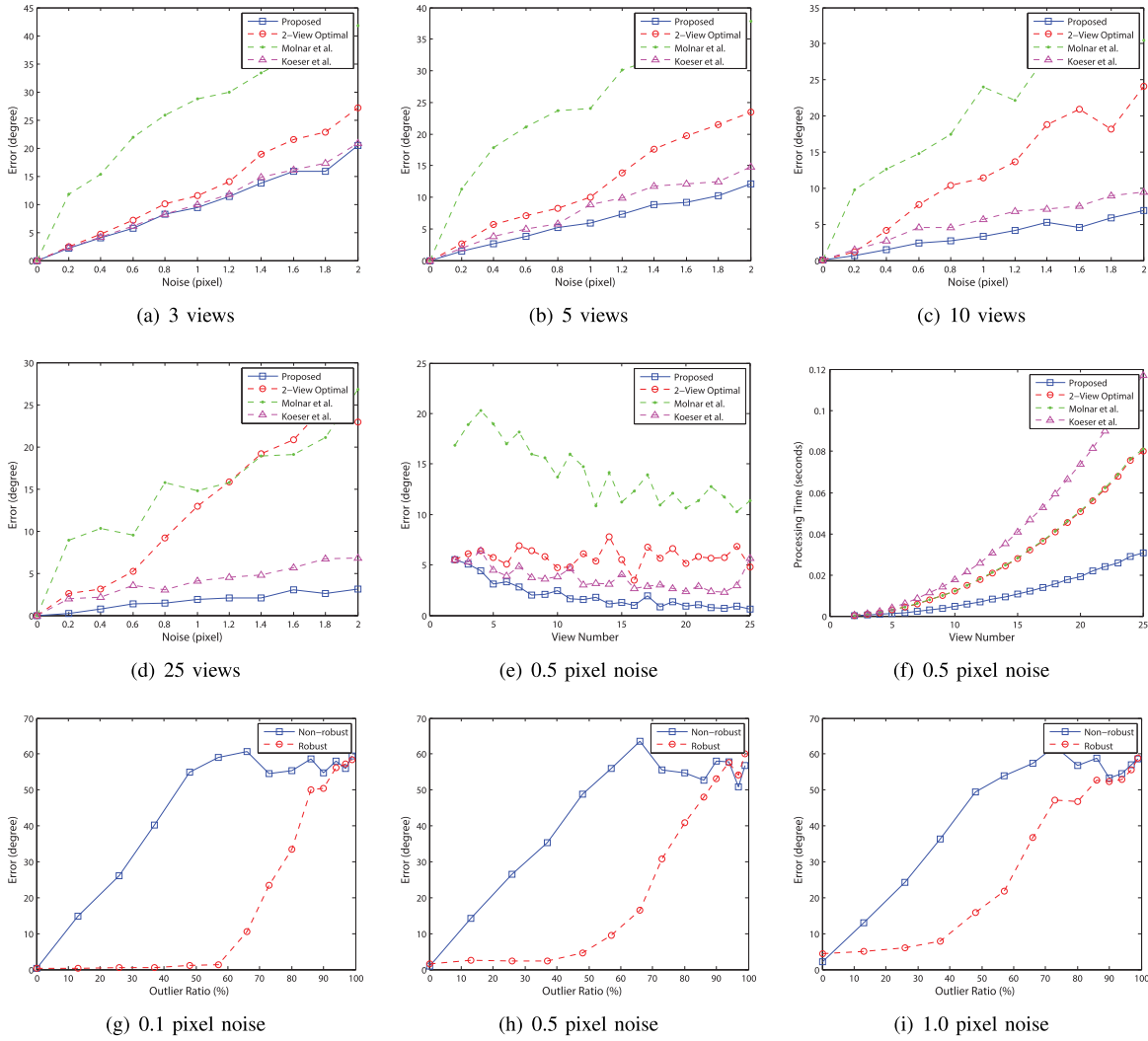


Fig. 4. Synthesized tests comparing normal estimators. (a-d) report the angular error plotted as the function of noise  $\sigma$  with different number of views; (e) and (f) are the error and the processing time w.r.t. increasing view number; (g-i) show the accuracy of the non-robust and robust algorithms w.r.t. increasing noise  $\sigma$  on different outlier levels.

TABLE I

SURFACE NORMAL ESTIMATION. FOR EACH METHOD, THE MEAN (AVG), MEDIAN (MED) ANGULAR ERRORS IN DEGREES, THE STANDARD DEVIATION, ( $\sigma$ ) AND THE PROCESSING TIME (T) GIVEN IN MILLISECONDS ARE REPORTED. TESTS (ROWS): (1) FOUNTAIN-P11, (2) HERZ-JESUS-P8, (3) HERZ-JESUS-P25 ARE FROM [23], (4) BOOKS1, (5) BOOKS2, (6) BAG ARE FROM [24] AND, FINALLY, (7) COURTYARD (8) DELIVERY AREA (9) PIPES (10) PLAYGROUND, (11) RELIEF AND (12) TERRACE ARE FROM ETH3D [25]

	FNE				Köser				2-Opt				MV-Opt				Robust MV-Opt			
	AVG	MED	$\sigma$	T	AVG	MED	$\sigma$	T	AVG	MED	$\sigma$	T	AVG	MED	$\sigma$	T	AVG	MED	$\sigma$	T
(1)	13.2	7.2	15.3	0.08	13.3	7.3	20.2	0.21	13.3	7.2	15.2	0.04	13.1	7.2	15.1	<b>0.01</b>	<b>5.7</b>	<b>4.4</b>	<b>5.4</b>	0.44
(2)	42.1	39.1	24.6	0.40	24.4	19.2	18.9	1.10	24.4	19.2	18.8	0.02	24.2	18.8	18.8	<b>0.01</b>	<b>3.4</b>	<b>13.3</b>	<b>0.6</b>	0.05
(3)	22.4	16.9	18.9	0.30	22.3	17.2	18.6	0.80	22.3	17.2	18.6	0.10	22.3	17.1	18.5	<b>0.03</b>	<b>9.6</b>	<b>4.4</b>	<b>12.2</b>	0.10
(4)	10.6	6.0	13.6	0.10	10.6	6.3	13.2	0.30	10.6	6.2	13.2	0.06	10.4	6.2	13.2	<b>0.02</b>	<b>5.9</b>	<b>4.3</b>	<b>7.8</b>	0.05
(5)	15.4	7.6	20.9	0.10	15.6	8.0	20.8	0.20	15.5	7.8	20.8	0.05	15.2	7.7	20.8	<b>0.01</b>	<b>11.4</b>	<b>5.5</b>	<b>19.7</b>	0.04
(6)	25.1	21.9	16.1	0.05	24.6	21.3	16.0	0.10	24.6	21.4	15.9	0.03	24.3	21.0	15.7	<b>0.01</b>	<b>18.9</b>	<b>16.6</b>	<b>12.1</b>	0.07
(7)	24.3	19.5	19.8	0.06	24.4	19.6	19.8	0.15	24.4	19.6	19.8	0.03	24.2	19.6	19.4	<b>0.01</b>	<b>12.3</b>	<b>9.0</b>	<b>11.0</b>	0.22
(8)	36.1	31.7	25.0	0.04	35.6	31.2	25.2	0.10	35.6	31.2	25.2	0.02	35.8	31.3	25.2	<b>0.01</b>	<b>18.3</b>	<b>11.4</b>	<b>19.0</b>	0.72
(9)	39.9	37.1	24.6	0.02	40.5	37.5	24.7	0.05	40.5	37.5	24.7	0.01	40.1	37.1	24.6	<b>0.01</b>	<b>20.3</b>	<b>12.7</b>	<b>21.8</b>	0.62
(10)	49.4	50.2	24.1	0.02	48.6	48.9	24.2	0.05	48.6	48.9	24.2	0.01	48.3	48.7	24.2	<b>0.01</b>	<b>36.0</b>	<b>29.7</b>	<b>24.7</b>	0.71
(11)	35.4	32.1	20.4	0.05	35.1	31.8	20.3	0.14	35.1	31.8	20.3	0.03	35.1	31.8	20.2	<b>0.01</b>	<b>29.4</b>	<b>27.3</b>	<b>16.8</b>	0.45
(12)	52.6	52.9	23.9	0.02	55.3	60.8	24.0	0.05	55.3	60.8	24.0	0.01	54.3	55.0	23.3	<b>0.01</b>	<b>39.2</b>	<b>34.2</b>	<b>22.9</b>	0.68
AVG	30.5	26.9	20.6	0.10	29.2	25.8	20.5	0.27	29.2	25.7	20.1	0.03	28.9	25.1	19.9	<b>0.01</b>	<b>17.5</b>	<b>14.4</b>	<b>14.5</b>	0.35
MED	30.3	26.8	20.7	0.06	24.5	20.5	20.3	0.15	24.5	20.5	20.1	0.03	24.3	20.3	19.8	<b>0.01</b>	<b>15.3</b>	<b>12.1</b>	<b>14.5</b>	0.33

shape of the features. The relative rotation between a pair of matched regions was defined by the dominant gradient directions. In the rest of the evaluation we applied TBMR.

The competitor algorithms are FNE [15], the method of Kevin Köser [11], the two-view optimal method (2-Opt), the proposed multi-view algorithm (MV-Opt) and its robust

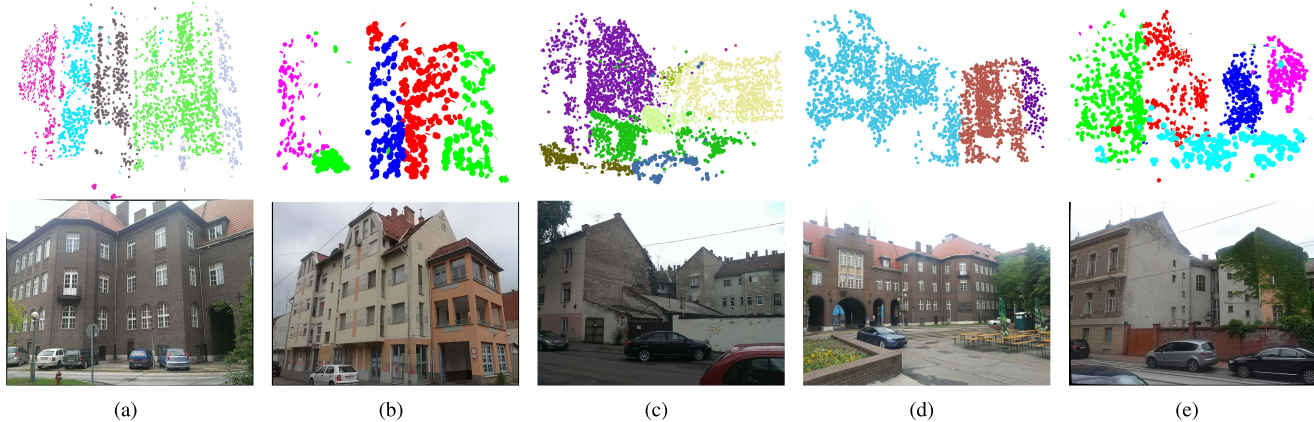


Fig. 5. Multi-plane fitting results. First row shows obtained 3D point cloud. Colors denote planes. Second row consists of an image of each sequence.

variant (Robust MV-Opt). Table I reports the results of the methods on each test scene (rows). Every block, consisting of four columns, shows the average (AVG) and median (MED) angular errors, their standard deviation ( $\sigma$ ), and the mean processing time of the point-wise computation in milliseconds. The mean and median results on all scenes are reported in the last two rows.

It can be seen that the optimal method without robust estimation (MV-Opt) is more accurate except two cases and, on average, one order of magnitude faster than the competitor algorithms. Even though its errors are the lowest, the difference is not significant, approx. 0.3 degrees. Since the synthesized tests reported larger difference, this means that the outlier percentage is high. Overcoming this problem, the robust algorithm (Robust MV-Opt) obtains twice as accurate surface normals with similar speed as the competitor methods. In Fig. 6, each row shows the result on a test sequence. The first column is an image from the sequence. The second and third ones show the reconstructed oriented point cloud rendered from different viewpoints.

In practice, the robust algorithm rejects  $\approx 60\%$  of the detected points. For the kept ones, the ratio of the view-pairs considered as inlier is  $\approx 70\%$  on average.

### C. Application: Improving PMVS2

In this section, we show that combining the proposed normal estimation technique with the state-of-the-art PMVS2 [27] structure-from-motion algorithm is beneficial and leads to superior results. PMVS2 has an initial seed point generation step applied before the dense reconstruction. During this step, it detects feature points and estimates surface normals applying an iterative strategy which minimizes a photo-consistency-based cost function. To demonstrate the accuracy of the proposed method, we replaced this normal estimation step with the proposed one. Each row of Table II is a test sequence. The first block, consisting of four columns, shows the error of the original PMVS2 w.r.t. the ground truth point cloud obtained by a laser scanner. The second block reports the results of PMVS2 combined with the proposed approach. The reported properties are: the mean error of the point cloud ( $\mathcal{E}_p$ , Euclidian distance), its standard deviation ( $\sigma_p$ ), the

TABLE II

THE ACCURACY OF THE ORIENTED POINT CLOUDS OBTAINED BY APPLYING THE ORIGINAL PMVS2 AND THE ONE COMBINED WITH THE PROPOSED NORMAL ESTIMATION.  $\mathcal{E}_p$  IS THE MEAN DISTANCE OF THE RECONSTRUCTED AND THE GROUND TRUTH POINTS AND  $\sigma_p$  IS THE STANDARD DEVIATION.  $\mathcal{E}_n$  IS THE MEAN ANGULAR ERROR (IN DEGREES) OF THE OBTAINED NORMALS W.R.T. THE GROUND TRUTH ONES,  $\sigma_n$  IS THE STANDARD DEVIATION OF THE ERRORS. TESTS (ROWS): (1) FOUNTAIN-P11, (2) HERZ-JESUS-P8, (3) HERZ-JESUS-P25 ARE FROM [23], (4) BOOKS1, (5) BOOKS2, (6) BAG ARE FROM [24]

	PMVS2				PMVS2 + Robust MV-Opt			
	$\mathcal{E}_p$	$\sigma_p$	$\mathcal{E}_n$	$\sigma_n$	$\mathcal{E}_p$	$\sigma_p$	$\mathcal{E}_n$	$\sigma_n$
(1)	0.013	0.015	25.6	19.1	<b>0.008</b>	<b>0.011</b>	<b>23.1</b>	<b>18.1</b>
(2)	0.077	0.052	33.2	22.7	<b>0.013</b>	<b>0.018</b>	<b>24.4</b>	<b>18.7</b>
(3)	0.023	0.028	27.6	19.8	<b>0.016</b>	<b>0.022</b>	<b>23.7</b>	<b>17.6</b>
(4)	<b>0.031</b>	<b>0.048</b>	<b>27.8</b>	<b>19.7</b>	0.032	0.051	28.1	<b>19.7</b>
(5)	0.057	0.063	32.0	20.6	<b>0.053</b>	<b>0.060</b>	<b>31.3</b>	<b>20.1</b>
(6)	0.050	<b>0.050</b>	31.8	18.5	<b>0.049</b>	<b>0.050</b>	<b>31.5</b>	<b>18.3</b>
AVG	0.042	0.043	29.7	20.1	<b>0.029</b>	<b>0.035</b>	<b>27.0</b>	<b>18.8</b>
MED	0.041	0.049	29.8	19.8	<b>0.024</b>	<b>0.036</b>	<b>26.2</b>	<b>18.5</b>

angular error of the normals ( $\mathcal{E}_n$ , in degrees) and, finally, its standard deviation ( $\sigma_n$ ). It can be seen that combining the proposed estimation technique with PMVS2 leads to more accurate reconstructions both in terms of the quality of the dense point cloud and that of the surface normals.

### D. Application: Plane Fitting

In this section, we demonstrate an application as the fitting of planes to an oriented point cloud obtained by the proposed technique. The objective of this section is to show that multi-model fitting algorithms are sensitive to the minimal method they use, *e.g.* fitting a plane to three points, to estimate the model hypotheses. Therefore, it is more beneficial to fit a plane to an oriented point, *i.e.* point with normal, than to three non-oriented ones.

We took several photos of buildings having large flat walls, then points are detected by ASIFT and the whole system is calibrated using OpenMVG [28] with a priori intrinsic camera parameters. Points are assigned manually to planes or the



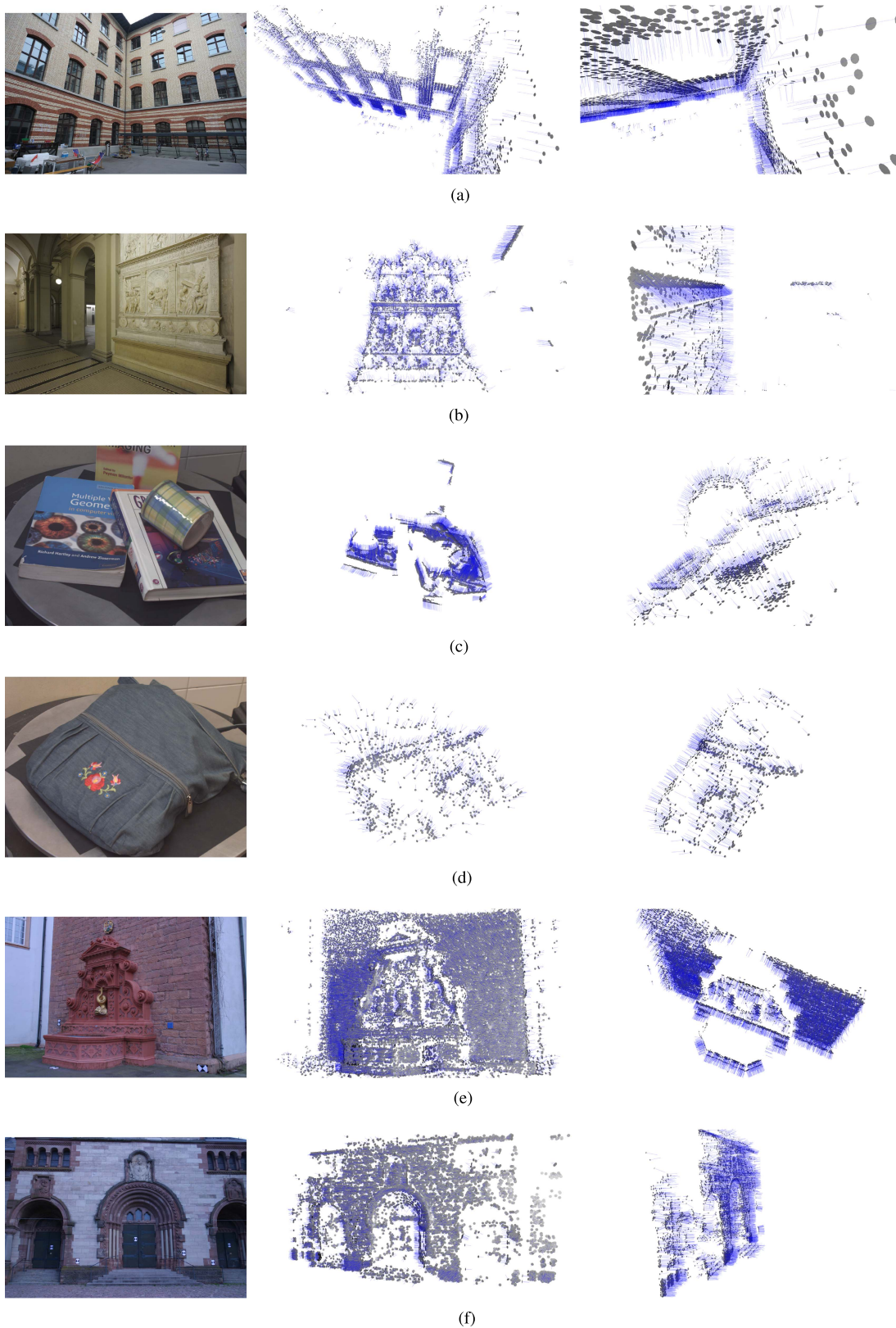


Fig. 6. Example results from each dataset. The first column is an image from the sequence, the remaining ones show the estimated normals (blue lines) and the triangulated points (gray patches) from different view-points. (a) courtyard (ETH3D). (b) relief (ETH3D). (c) books2 (Pusztai). (d) bag (Pusztai). (e) fountain-p11 (Strecha). (f) herz-jesus-p25 (Strecha).

outlier class, *i.e.* points not belonging to any dominant planes, to have a ground truth clustering. The properties of each scene is written in Table IV. We chose PEARL [3], T-linkage [29]

and MFIGP [30] algorithms for multi-model fitting since they have publicly available source codes and can be considered as state-of-the-art techniques.



TABLE III

MULTIPLE PLANE FITTING TO ORIENTED (1PT) AND NON-ORIENTED (3PT) POINT CLOUDS USING STATE-OF-THE-ART MULTI-MODEL FITTING ALGORITHMS (EACH PAIR OF ROWS). THE SURFACE NORMALS ARE OBTAINED BY THE PROPOSED METHOD. THE MEAN MISCLASSIFICATION ERROR IN PERCENTAGE IS REPORTED FOR EACH TEST CASE (COLUMNS; 1 CORRESPONDS TO FIG. 5). THE PROPERTIES OF EACH SCENE ARE WRITTEN IN TABLE IV

		(a)	(b)	(c)	(d)	(e)	AVG
PEARL [3]	1PT	<b>12</b>	<b>23</b>	<b>37</b>	39	<b>10</b>	<b>24</b>
	3PT	16	31	40	<b>36</b>	11	27
T-link [29]	1PT	<b>16</b>	39	<b>39</b>	<b>57</b>	<b>17</b>	<b>34</b>
	3PT	20	<b>33</b>	44	<b>57</b>	27	36
MFIGP [30]	1PT	<b>50</b>	59	<b>52</b>	<b>52</b>	<b>51</b>	<b>52</b>
	3PT	56	<b>46</b>	56	57	59	56

TABLE IV

THE PROPERTIES OF MULTI-PLANE FITTING SCENES. THE POINT NUMBER (1ST ROW), PLANE NUMBER (2ND ROW) AND OUTLIER PERCENTAGE (3RD ROW) ARE REPORTED FOR EACH TEST CASE (COLUMNS, CORRESPONDS TO FIG. 5). THE CLUSTERING RESULTS ARE IN TABLE III

	(a)	(b)	(c)	(d)	(e)
Point #	3 257	2 105	4 391	2 758	1 749
Plane #	6	6	8	6	5
Outlier %	16%	15%	31%	11%	21%

Table III reports the clustering results of each column in Fig. 5. The first row of the table denotes the test case. The second and third rows show the results of PEARL generating the initial model-hypotheses exploiting the surface normals (1PT) or not (3PT), respectively. The error is the misclassification error (ME), *i.e.* the ratio of the misclassified points:

$$ME = \frac{\#\text{Misclassified Points}}{\#\text{Points}}$$

It can be seen that applying PEARL to oriented point clouds leads to the most accurate results in all but one case.

## V. CONCLUSION

In this paper, an optimal method is proposed for two-view surface normal estimation, then it is extended to multiple views. The method estimates a normal for each affine correspondence individually, and its robust version is able to deal with approx. 60-70% outlier ratio. It is superior to the state-of-the-art both in synthesized tests and on publicly available real datasets. Comparing with other components of a structure-from-motion pipeline, the technique has negligible time demand despite the pair-wise term since the coefficient computation is efficient and only the obtained polynomial equation has to be solved. Usually limited number of views are given, at most 10–20, where a point can be tracked. Therefore, it is very rare to have problems for which the computation lasts even for a few milliseconds. In our C++ implementation the processing time of 100 views is  $\approx 7$  milliseconds. However, aiming at real time capability for thousands of point

sequences, both the coefficient calculation for each view and the processing of each point sequence can be parallelized and implemented on GPU straightforwardly. Using the obtained oriented point cloud in PMVS or multi-plane fitting applications is beneficial and leads to significant improvement in accuracy as it is demonstrated experimentally.

## APPENDIX

### LOCAL AFFINITIES FOR ARBITRARY CAMERA MODEL

Suppose that a 3D point  $\mathbf{P} = [x \ y \ z]^T$  lying on a continuous surface  $S$  is given by its projections in two images  $\mathbf{p}_1 = [u_1 \ v_1]^T$  and  $\mathbf{p}_2 = [u_2 \ v_2]^T$ . For the  $i$ th image, the projected coordinates  $u_i$  and  $v_i$  are determined by the projective functions  $u_i = \Pi_x^i(x, y, z)$ ,  $v_i = \Pi_v^i(x, y, z)$ , where  $S(u \ v) = [x \ y \ z]^T$  is written in parametric form as  $x = \mathcal{X}(u, v)$ ,  $y = \mathcal{Y}(u, v)$ ,  $z = \mathcal{Z}(u, v)$ . It is well-known in differential geometry [31] that the basis of the tangent plane at point  $\mathbf{P}$  is written by the partial derivatives of  $S$  w.r.t. the spatial coordinates. The surface normal  $\mathbf{n}$  is expressed by the cross product of the tangent vectors  $\mathbf{s}_u$  and  $\mathbf{s}_v$  where  $\mathbf{s}_u = \left[ \frac{\partial \mathcal{X}(u, v)}{\partial u} \ \frac{\partial \mathcal{Y}(u, v)}{\partial u} \ \frac{\partial \mathcal{Z}(u, v)}{\partial u} \right]^T$ , and  $\mathbf{s}_v = \left[ \frac{\partial \mathcal{X}(u, v)}{\partial v} \ \frac{\partial \mathcal{Y}(u, v)}{\partial v} \ \frac{\partial \mathcal{Z}(u, v)}{\partial v} \right]^T$ . Finally,  $\mathbf{n} = \mathbf{s}_u \times \mathbf{s}_v$ . Locally, around point  $\mathbf{P}$ , the surface can be approximated by the tangent plane, therefore, the neighboring points in the  $i$ th image are written as the first-order Taylor-series as follows:

$$\mathbf{p}_i + \Delta = \begin{bmatrix} x_i + \Delta x \\ y_i + \Delta y \end{bmatrix} \approx \begin{bmatrix} \Pi_x(x, y, z) \\ \Pi_y(x, y, z) \end{bmatrix} + \begin{bmatrix} \frac{\partial \Pi_x^i(x, y, z)}{\partial u} & \frac{\partial \Pi_x^i(x, y, z)}{\partial v} \\ \frac{\partial \Pi_y^i(x, y, z)}{\partial u} & \frac{\partial \Pi_y^i(x, y, z)}{\partial v} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix},$$

where  $[\Delta v \ \Delta u]^T$  is the translation on surface  $S$ , and  $\Delta x, \Delta y$  are the coordinates of the implied translation added to  $\mathbf{p}_i$ . It can be seen that transformation  $\mathbf{J}_i$  mapping the infinitely close vicinity around point  $\mathbf{p}_i$  in the  $i$ th image is given as

$$\mathbf{J}_i = \begin{bmatrix} \frac{\partial \Pi_x^i(x, y, z)}{\partial u} & \frac{\partial \Pi_x^i(x, y, z)}{\partial v} \\ \frac{\partial \Pi_y^i(x, y, z)}{\partial u} & \frac{\partial \Pi_y^i(x, y, z)}{\partial v} \end{bmatrix},$$

thus  $[\Delta x \ \Delta y]^T \approx \mathbf{J}_i [\Delta u \ \Delta v]^T$ . The partial derivatives are reformulated using the chain rule. As an example, the first element it is as

$$\frac{\partial \Pi_x^i(x, y, z)}{\partial u} = \frac{\partial \Pi_x^i(x, y, z)}{\partial x} \frac{x}{\partial u} + \frac{\partial \Pi_x^i(x, y, z)}{\partial x} \frac{y}{\partial x} + \frac{\partial \Pi_x^i(x, y, z)}{\partial x} \frac{z}{\partial x} = \nabla(\Pi_x^i)^T \mathbf{s}_u,$$

where  $\nabla \Pi_x^i$  is the gradient vector of  $\Pi_x$  w.r.t. coordinates  $x, y$  and  $z$ . Similarly,

$$\frac{\partial \Pi_x^i}{\partial v} = \nabla(\Pi_x^i)^T \mathbf{s}_v, \quad \frac{\partial \Pi_y^i}{\partial u} = \nabla(\Pi_y^i)^T \mathbf{s}_u, \quad \frac{\partial \Pi_y^i}{\partial v} = \nabla(\Pi_y^i)^T \mathbf{s}_v,$$

Therefore,  $\mathbf{J}_i$  can be written as

$$\mathbf{J}_i = \begin{bmatrix} \nabla(\Pi_x^i)^T \\ \nabla(\Pi_y^i)^T \end{bmatrix} [\mathbf{s}_u \ \mathbf{s}_v].$$

Affine transformation  $\mathbf{A}$  transforming the infinitely close vicinity of point  $\mathbf{p}_1$  in the 1st image to that of  $\mathbf{p}_2$  in the 2nd one is as follows:  $[\Delta x_2 \ \Delta y_2]^T = \mathbf{J}_2 \mathbf{J}_1^{-1} [\Delta x_1 \ \Delta y_1]^T = \mathbf{A} [\Delta x_1 \ \Delta y_1]^T$ .

*Surface Normals:* Using the well-known formula  $\mathbf{s}_b \mathbf{s}_u^T - \mathbf{s}_u \mathbf{s}_b^T = [\mathbf{n}]_{\times}$ , where  $[\cdot]_{\times}$  is the cross-product matrix, local affinity  $\mathbf{A} = \mathbf{J}_2 \mathbf{J}_1^{-1}$  is written as

$$\mathbf{A} = \frac{1}{(\Pi_x^1)^T [\mathbf{n}]_{\times} \Pi_y^1} \begin{bmatrix} (\Pi_x^2)^T [\mathbf{n}]_{\times} \Pi_y^1 & (\Pi_x^1)^T [\mathbf{n}]_{\times} \Pi_x^2 \\ (\Pi_y^2)^T [\mathbf{n}]_{\times} \Pi_y^1 & (\Pi_x^1)^T [\mathbf{n}]_{\times} \Pi_y^2 \end{bmatrix}.$$

Formula  $\mathbf{a}^T [\mathbf{n}]_{\times} \mathbf{b}$  is called the scalar triple product, it equals to  $\mathbf{n}^T (\mathbf{b} \times \mathbf{a})$ . Therefore, the final formula for the local affine transformation is written as

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} = \frac{1}{\mathbf{n}^T \mathbf{w}_5} \begin{bmatrix} \mathbf{n}^T \mathbf{w}_1 & \mathbf{n}^T \mathbf{w}_2 \\ \mathbf{n}^T \mathbf{w}_3 & \mathbf{n}^T \mathbf{w}_4 \end{bmatrix}, \quad (17)$$

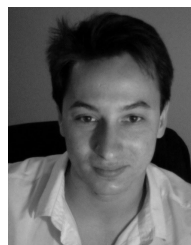
where

$$\begin{aligned} \mathbf{w}_1 &= \nabla \Pi_y^1 \times \nabla \Pi_x^2 & \mathbf{w}_2 &= \nabla \Pi_x^2 \times \nabla \Pi_x^1 \\ \mathbf{w}_3 &= \nabla \Pi_y^1 \times \nabla \Pi_y^2 & \mathbf{w}_4 &= \nabla \Pi_y^2 \times \nabla \Pi_x^1 \\ \mathbf{w}_5 &= \nabla \Pi_y^1 \times \nabla \Pi_x^1. \end{aligned}$$

Eq. 17 describes the relationship of surface normals and local affinities. Since the constraint is to have a projection function differentiable around the observed 3D point, the relationship holds for *general camera models*.

## REFERENCES

- [1] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Eurograph. Symp. Geometry Process.*, 2006, pp. 61–70.
- [2] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, p. 29, 2013.
- [3] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *Int. J. Comput. Vis.*, vol. 97, no. 2, pp. 123–147, 2012.
- [4] R. J. Woodham, "Photometric method for determining surface orientation from multiple images," *Opt. Eng.*, vol. 19, no. 1, pp. 139–144, 1980.
- [5] F. E. Nicodemus, "Directional reflectance and emissivity of an opaque surface," *Appl. Opt.*, vol. 4, no. 7, pp. 767–775, 1965.
- [6] C. H. Esteban, G. Vogiatzis, and R. Cipolla, "Multiview photometric stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 548–554, Mar. 2008.
- [7] Y. Quéau, R. Mecca, J.-D. Durou, and X. Descombes, "Photometric stereo with only two images: A theoretical study and numerical resolution," *Image Vis. Comput.*, vol. 57, pp. 175–191, Jan. 2017.
- [8] O. D. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 2, no. 3, pp. 485–508, 1988.
- [9] E. Malis and M. Vargas, "Deeper understanding of the homography decomposition for vision-based control," INRIA, Res. Rep. RR-6303, 2007, p. 90. [Online]. Available: <https://hal.inria.fr/inria-00174036>
- [10] H. Liu, "Deeper understanding on solution ambiguity in estimating 3D motion parameters by homography decomposition and its improvement," Ph.D. dissertation, Dept. Fiber Amenity Eng., University of Fukui, Fukui Prefecture, Japan, 2012.
- [11] K. Köser, "Geometric estimation with local affine frames and free-form surfaces," Ph.D. dissertation, Dept. Faculty Eng., Kiel Univ., Kiel, Germany, 2009.
- [12] K. Mikolajczyk *et al.*, "A comparison of affine region detectors," *Int. J. Comput. Vis.*, vol. 65, pp. 43–72, Nov. 2005.
- [13] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 438–469, 2009.
- [14] D. Mishkin, J. Matas, and M. Perdoch, "MODS: Fast and robust method for two-view matching," *Comput. Vis. Image Understand.*, vol. 141, pp. 81–93, Dec. 2015.
- [15] D. Barath, J. Molnar, and L. Hajder, "Novel methods for estimating surface normals from affine transformations," in *Proc. Revised Sel. Papers (VISAPP)*, 2015, pp. 316–337.
- [16] Y. Furukawa and J. Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [17] Y. Furukawa and J. Ponce, *Patch-based Multi-view Stereo Software*. Accessed: Jul. 17, 2014. [Online]. Available: <http://www.di.ens.fr/pmvs>
- [18] P. F. Georgel, S. Benhimane, and N. Navab, "A unified approach combining photometric and geometric information for pose estimation," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 1–10.
- [19] D. Barath, J. Matas, and L. Hajder, "Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry," in *Proc. BMVC*, 2016, pp. 1–12.
- [20] M. Clegg, J. Edmonds, and R. Impagliazzo, "Using the Groebner basis algorithm to find proofs of unsatisfiability," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 174–183.
- [21] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 138–156, 2000.
- [22] Å. Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA, USA: SIAM, 1996.
- [23] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. CVPR*, Jun. 2008, pp. 1–8.
- [24] Z. Pustai and L. Hajder, "Ground-truth tracking data generation using rotating real-world objects," in *Proc. Revised Sel. Papers (VISAPP)*, 2017, pp. 395–417.
- [25] T. Schöps *et al.*, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Proc. CVPR*, Jul. 2017, pp. 3260–3269.
- [26] Y. Xu, P. Monasse, and T. Géraud, and L. Najman, "Tree-based morse regions: A topological approach to local feature detection," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5612–5625, Dec. 2014.
- [27] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2008.
- [28] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *Proc. Int. Workshop Reproducible Res. Pattern Recognit.*, 2016, pp. 60–74.
- [29] L. Magri and A. Fusiello, "T-linkage: A continuous relaxation of j-linkage for multi-model fitting," in *Proc. CVPR*, Jun. 2014, pp. 3954–3961.
- [30] T. T. Pham, T.-J. Chin, K. Schindler, and D. Suter, "Interacting geometric priors for robust multimodel fitting," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4601–4610, Oct. 2014.
- [31] E. Kreyszig, *Introduction to Differential Geometry and Riemannian Geometry*, vol. 16. Toronto, Ontario, Canada: University of Toronto Press, 1968.



**Dániel Baráth** was born in Budapest, Hungary, in 1989. He is currently pursuing the Ph.D. degree with Eötvös Loránd University. He is also a member of the Machine Perception Research Laboratory, Institute for Computer Science and Control, MTA SZTAKI and the Centre for Machine Perception, Czech Technical University, Prague. His research interests are minimal methods in computer vision and robust model estimation.



**Ivan Eichhardt** received the M.Sc. degree in computer science from Eötvös Loránd University, Budapest, Hungary, in 2014, where he is currently pursuing the degree with the Doctoral School of Informatics. He is also a member of the Machine Perception Research Laboratory, Institute for Computer Science and Control, MTA SZTAKI. His main research fields are computer vision and image processing, including structure from motion, surface reconstruction, and sensor data processing and fusion.



**Levente Hajder** was born in Budapest, Hungary, in 1975. He received the degree in computer science from Kandó Kálmán Polytechnics and the degree in electrical engineering from the Budapest University of Technology and Economics, and the Ph.D. degree in 2008. He is currently an Associate Professor with the Department of Algorithms and Their Applications, Eötvös Loránd University, Budapest. His research interests are optimization methods in computer vision and 3D reconstruction.