Efficient Recovery of Essential Matrix from Two Affine Correspondences

Daniel Barath, and Levente Hajder

***Abstract*—We propose a method to estimate the essential matrix using two affine correspondences for a pair of calibrated perspective cameras. Two novel, linear constraints are derived between the essential matrix and a local affine transformation. The proposed method is also applicable to the over-determined case. We extend the normalization technique of Hartley to local affinities and show how the intrinsic camera matrices modifies them. Even though perspective cameras are assumed, the constraints can straightforwardly be generalized to arbitrary camera models since they describe the relationship between local affinities and epipolar lines (or curves). Benefiting from the low number of exploited points, it can be used in robust estimators, e.g. RANSAC, as an engine, thus leading to significantly less iterations than the traditional point-based methods. The algorithm is validated both on synthetic and publicly available datasets and compared with the state-of-the-art. Its applicability is demonstrated on two-view multi-motion fitting, i.e. finding multiple fundamental matrices simultaneously, and outlier rejection.**

***Index Terms*—epipolar geometry, essential matrix, affine correspondence, minimal method**

I. Introduction

The estimation of epipolar geometry between a pair of images is a key-problem for the recovery of relative camera motion and has been studied for decades. Luong and Fougeras showed that this relationship can be described by the so-called 3×3 fundamental matrix [1]. Since then, several approaches have been proposed to cope with this problem. The well-known seven and eight-point algorithms [2] need no a priori information about the camera parameters to estimate the fundamental matrix from point correspondences. However, exploiting the intrinsic camera parameters (focal length, principal point, etc.), the estimation can be done using six [3], [4], [5], [6] or five correspondences [7], [8], [9], [10].

In this paper, we assume intrinsic parameters and two affine correspondences to be known between a pair of images to recover the essential matrix. An affine correspondence consists of a point pair and the related local affine transformation mapping the infinitesimally close vicinity of the point in the first image to that of in the second one. Nowadays, several approaches are available for the estimation of local affine transformations. Beside the well-known affine-covariant feature detectors [11] such as MSER, Hessian-Affine, Harris-Affine, there are some modern ones based on view-synthesizing, e.g. ASIFT [12], ASURF or MODS [13]. They obtain accurate local affinities and many correspondences by transforming the original image with an affine transformation to create a synthetic view. Then a feature detector is applied to the warped images. The final local affinity related to a point pair is estimated as the combination of the transformation regarding to the current synthetic view and the affine transformation which the applied detector obtains.

Using local affinities for fundamental matrix estimation is not a new idea. Perdoch et al. [14] and Chum et al. [15] proposed methods using two and three affine correspondences, respectively. Even so, they provide only approximations – the error is not zero even for noise-free input – since they generate point correspondences exploiting local affine transformations and apply the six [3] and eight-point algorithms [2], respectively. Nevertheless, local affinities cannot generate point correspondences since they are defined as the partial derivative, w.r.t. the image directions, of the related homography. Thereby, they are valid only infinitesimally close to the observed point [16]. Bentolila et al. [17] showed that two affine transformations yields three conic constraints on fundamental matrix estimation and three affine correspondences are enough. Recently, an approach is proposed by Raposo and Barreto [18] which is slightly similar to the base algorithm proposed in this paper. Providing a derivation on the basis of homographies and applying the solver of the five-point algorithm [8], they estimate the epipolar geometry using two affine correspondences. Unlike them, we show that this relationship can be formalized directly, considering the way how a local affinity affects the epipolar lines. Through the proposed formulation, it can straightforwardly be seen that the relationship holds for arbitrary central camera models. Also, the solver we propose leads to results superior to [18] as it is demonstrated in Sec. IV.

The contributions of this paper are as follows: **(i)** Two linear constraints are derived from a local affine transformation showing its direct relationship to the epipolar geometry – the way how it affects the epipolar lines. Not approaching the problem as a derivation of homographies (as [18] does), the constraints can easily be generalized to arbitrary camera model, e.g. omni-directional ones. **(ii)** The proposed constraints make the estimation possible using two affine correspondences. The method is generalized to solve the over-determined case as well and provides only one globally optimal essential matrix. It is demonstrated both on synthesized and real world test that the algorithm is superior to the state-of-the-art in term of the accuracy of the estimated camera motion. **(iii)** It is shown how the multiplication of the point

Daniel Barath and Levente Hajder were with the Machine Perception Research Laboratory, MTA SZTAKI, Budapest, 1111 Hungary. Daniel Barath were also with the Centre for Machine Perception, Department of Cybernetics Czech Technical University, Prague, Czech Republic. E-mail: {barath.daniel, hajder.levente}@sztaki.mta.hu.

Manuscript received July 19, 2017

locations by the camera matrices modifies the local affinities, thus making the method applicable to image pairs captured by different camera set ups. The normalization technique of Hartley [19] is extended to affine transformations to achieve numerically stable estimates in the over-determined case.

II. Preliminaries and Notation

2D point correspondences are represented by their homogeneous form as $\mathbf{p} = \begin{bmatrix} u & v & 1 \end{bmatrix}^{\mathrm{T}}$ (1st image) and $\mathbf{p}' = \begin{bmatrix} u' & v' & 1 \end{bmatrix}^{\mathrm{T}}$ (2nd image). The related local affine transformation $\mathbf{A}$ is written as its linear part (left 2×2 submatrix) since the translation is determined by the point locations:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}. \tag{1}$$

An affine correspondence (AC) consists of a point pair and the related local affinity.

Let operator $\mathbf{M}_{[i:k,j:l]}$ denote the $(k-i+1) \times (l-j+1)$-sized submatrix of matrix $\mathbf{M}$ ($0 < i < k$ and $0 < j < l$). Vector $\mathbf{v}_{[i:k]}$ is the vector consisting of the elements of vector $\mathbf{v}$ from ith to kth ($i < k$). Formula $|\mathbf{v}|$ is considered as the L_2 norm of $\mathbf{v}$.

The ith element of the essential and fundamental matrices ($\mathbf{E}$ and $\mathbf{F}$) in row-major order is denoted as e_i and f_i, respectively ($i \in [1, 9]$). In contrast to the rest of the paper, in the appendix, the elements of $\mathbf{F}$ are indexed as f_{jk} ($j, k \in [1, 3]$).

The relationship of essential and fundamental matrices is written as $\mathbf{F} = \mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-1}$, where $\mathbf{K}$ and $\mathbf{K}'$ are the intrinsic parameters of the two cameras. Fundamental matrix $\mathbf{F}$ ensures the epipolar constraint as $\mathbf{p}'^{\mathrm{T}}\mathbf{F}\mathbf{p} = \mathbf{p}'^{\mathrm{T}}\mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-1}\mathbf{p} = 0$. In the rest of the paper, we assume that points $\mathbf{p}$ and $\mathbf{p}'$ have been premultiplied by $\mathbf{K}$ and $\mathbf{K}'$. This assumption simplifies the epipolar constraint to

$$\mathbf{q}'^{\mathrm{T}}\mathbf{E}\mathbf{q} = 0, \tag{2}$$

where $\mathbf{q}$ and $\mathbf{q}'$ are the points multiplied by $\mathbf{K}$ and $\mathbf{K}'$. Two additional constraints can be considered on the essential matrix $\mathbf{E}$. The first one is called trace constraint [2], it is as follows:

$$2\mathbf{E}\mathbf{E}^{\mathrm{T}}\mathbf{E} - \mathrm{tr}(\mathbf{E}\mathbf{E}^{\mathrm{T}})\mathbf{E} = 0. \tag{3}$$

This matrix equation yields nine polynomial equations for the elements of $\mathbf{E}$. The second restriction ensures that the determinant of the essential matrix must be zero:

$$\det(\mathbf{E}) = 0. \tag{4}$$

These two properties will help us to recover the essential and fundamental matrices exploiting two affine correspondences.

III. Two-point Algorithm

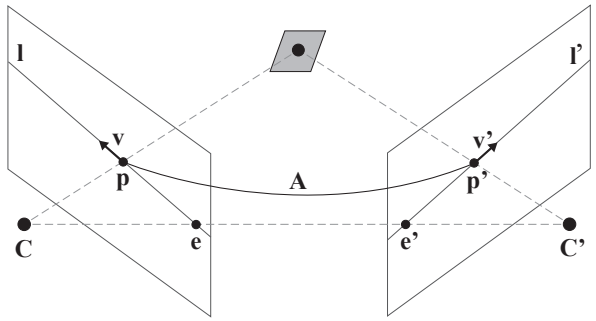
First, the linear relationship of the essential matrix and an affine transformation is described in this section. Then we exploit it to estimate the essential matrix from two affine correspondences.

A. Relationship of Essential Matrix and Local Affinities

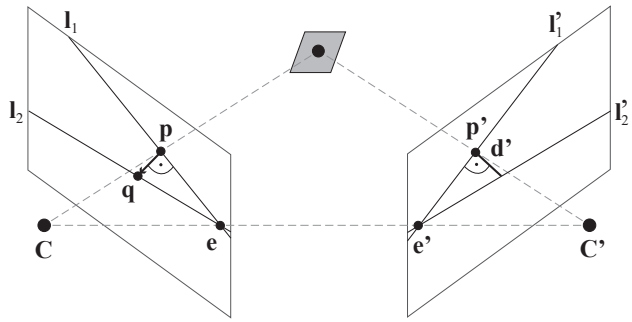
The aim of this section is to show the direct relationship of the essential matrix and a local affinity and to prove that it can be written in linear form. Even though we derive it for $\mathbf{E}$, these formulas hold for $\mathbf{F}$ if the point locations have not been (pre)multiplied by the intrinsic matrices. Local affine transformation $\mathbf{A}$ is defined as the partial derivative of the projection function [16]. Note that $\mathbf{A}$ has to be modified by the intrinsic matrices before the estimation, this will be shown in a latter section.

Suppose that essential matrix $\mathbf{E}$, point pair $\mathbf{p}$, $\mathbf{p}'$, and the related affinity $\mathbf{A}$ are given. It can be proven straightforwardly that $\mathbf{A}$ transforms $\mathbf{v}$ to $\mathbf{v}'$ (see Fig. 1(a)), where $\mathbf{v}$ and $\mathbf{v}'$ are the directions of the epipolar lines ($\mathbf{v}, \mathbf{v}' \in \mathbb{R}^2$) in the 1st and 2nd images [17], respectively. It can be seen that transforming the infinitesimally close vicinity of $\mathbf{p}$ to that of $\mathbf{p}'$, $\mathbf{A}$ has to map the lines going through the points. Therefore, $\mathbf{A}\mathbf{v} \parallel \mathbf{v}'$.

Note that this statement holds for arbitrary central camera models, e.g. omni-directional ones, since the line directions are determined by the first-order approximation, i.e. the local affinity, of the projection functions [20].



(a) Projections $\mathbf{p}$ and $\mathbf{p}'$ of a spatial point are given on cameras $\mathbf{C}$ and $\mathbf{C}'$. Vectors $\mathbf{v}$ and $\mathbf{v}'$ are the directions of the corresponding epipolar lines $\mathbf{l}$ and $\mathbf{l}'$. Local affine transformation $\mathbf{A}$ transforms $\mathbf{v}$ into $\mathbf{v}'$.



(b) The constraint for scale states that the ratio of $|p-q|$ and d' determines the scale between vectors $\mathbf{A}^{-\mathrm{T}}\mathbf{n}$ and $\mathbf{n}'$.

Fig. 1. The proposed constraints.

As it is well-known from computer graphics [21], formula $\mathbf{A}\mathbf{v} \parallel \mathbf{v}'$ can be reformulated as follows:

$$\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \beta\mathbf{n}', \tag{5}$$

where $\mathbf{n}$ and $\mathbf{n}'$ are the normals of the epipolar lines ($\mathbf{n}, \mathbf{n}' \in \mathbb{R}^2$, $\mathbf{n} \perp \mathbf{v}$, $\mathbf{n}' \perp \mathbf{v}'$). Scalar β denotes the scale between the transformed and the original vectors if $|\mathbf{n}| = 1$ and $|\mathbf{n}'| = 1$.

These normals are calculated as the first two coordinates of epipolar lines

$$\mathbf{l} = \mathbf{E}^{\mathrm{T}}\mathbf{p}' = \begin{bmatrix} a & b & c \end{bmatrix}^{\mathrm{T}}, \quad \mathbf{l}' = \mathbf{E}\mathbf{p} = \begin{bmatrix} a' & b' & c' \end{bmatrix}^{\mathrm{T}}. \quad (6)$$

Since the common scale regarding to normals $\mathbf{n} = \mathbf{l}_{[1:2]} = \begin{bmatrix} a & b \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{n}' = \mathbf{l}'_{[1:2]} = \begin{bmatrix} a' & b' \end{bmatrix}^{\mathrm{T}}$ is originated from the essential matrix, Eq. 5 is modified as follows:

$$\mathbf{A}^{-\mathrm{T}}\mathbf{n} = -\mathbf{n}'. \quad (7)$$

Detailed proof can be seen in the Appendix. Formulas 6 and 7 yield two equations which are linear in the parameters of the essential matrix as follows:

$$\begin{aligned}(u' + a_1 u)e_1 + a_1 v e_2 + a_1 e_3 + (v' + a_3 u)e_4 + \\ a_3 v e_5 + a_3 e_6 + e_7 = 0 \quad (8)\end{aligned}$$

$$\begin{aligned}a_2 u e_1 + (u' + a_2 v)e_2 + a_2 e_3 + a_4 u e_4 + \\ (v' + a_4 v)e_5 + a_4 e_6 + e_8 = 0. \quad (9)\end{aligned}$$

where a_i is the ith element of $\mathbf{A}$ in row-major order ($i \in [1, 4]$), as it is defined in Eq. 1. Points (u, v) and (u', v') are the points in the images, and e_j ($j \in [1, 9]$) is the jth element of the essential matrix

To summarize this section, *the linear part* of a local affine transformation *gives two equations*, represented by linear formulas, for essential matrix estimation. A point correspondence yields a third one through the epipolar constraint. Therefore an affine correspondence leads to three constraints. As the essential matrix has five Degrees-of-Freedom (DoF), two affine correspondences are enough for estimating $\mathbf{E}$, moreover, the estimation is overdetermined.

Remark that the proposed formulas (Eq.22) of [18] are exactly the same. The solution in [18] however is found intuitively after algebraic manipulations with no geometric interpretation provided. In contrast, we proved here that these formulas describe the way how a local affinity transforms the normal of the epipolar lines between the images.

B. The Proposed Solver

In this section, the proposed 2-point algorithm based on the introduced constraints is discussed. Suppose that two point pairs $(\mathbf{p}_1, \mathbf{p}'_1)$ and $(\mathbf{p}_2, \mathbf{p}'_2)$ and the related affinities $\mathbf{A}_1$ and $\mathbf{A}_2$ are given. Fig. 2 shows how $\mathbf{A}_1$ and $\mathbf{A}_2$ transform the infinitesimally close vicinities of the points from the first to the second images.

For the ith ($i \in \{1, 2\}$) correspondence, the combination of formulas Eqs. 8, 9, and Eq. 2 can be written as $\mathbf{C}_i\mathbf{x} = 0$, where $\mathbf{x} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{bmatrix}^{\mathrm{T}}$ is the vector of the unknown elements of the essential matrix. Matrix $\mathbf{C}_i$ is the coefficient matrix consisting of three rows, where the first two are the coefficients of Eqs. 8, 9. The third one contains the coefficients related to the well-known formula $\mathbf{p}'^{\mathrm{T}}\mathbf{E}\mathbf{p} = 0$. Note that the algorithm can straightforwardly be extended to $n > 2$ points by concatenating their $\mathbf{C}_i$ matrices. If at least three correspondences are given, the solution vector $\mathbf{x}$ is obtained as the eigenvector related to the smallest eigenvalue of matrix $\mathbf{C}^{\mathrm{T}}\mathbf{C}$, where matrix $\mathbf{C}$ is the concatenated coefficient matrix and of size $3n \times 9$.

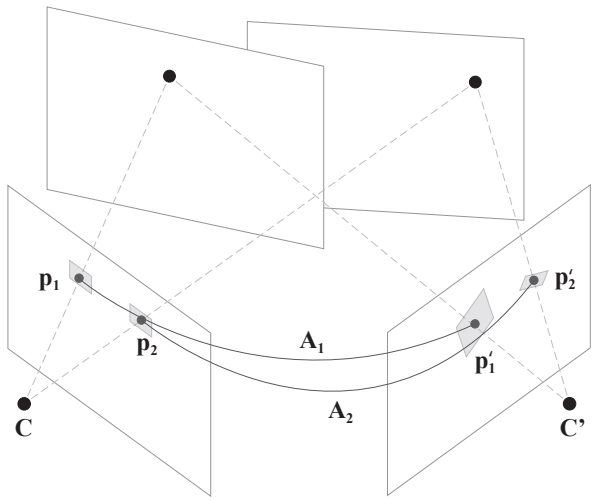


Fig. 2. Projections of two spatial points are given on cameras $\mathbf{C}$ and $\mathbf{C}'$. Corresponding local affine transformations $\mathbf{A}_1$ and $\mathbf{A}_2$ transforms the infinitesimally close vicinities of point pairs $(\mathbf{p}_1, \mathbf{p}'_1)$ and $(\mathbf{p}_2, \mathbf{p}'_2)$ between the image pair.

Considering the two point case, $\mathbf{C}$ is of size 6×9 as $\mathbf{C} = \begin{bmatrix} \mathbf{C}_1^{\mathrm{T}} & \mathbf{C}_2^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. Its null space is 3-dimensional, therefore, the solution of the system is given by the linear combination of the three corresponding singular vectors of $\mathbf{C}$ as

$$\mathbf{x} = \alpha\mathbf{d} + \beta\mathbf{e} + \gamma\mathbf{f}, \quad (10)$$

where $\mathbf{d}$, $\mathbf{e}$, and $\mathbf{f}$ are the singular vectors. Parameters α, β, and γ are unknown non-zero scalar values. These scalars are defined up to a common scale, therefore, one of them can be chosen to an arbitrary value. In the proposed algorithm, $\gamma = 1$.

By substituting this formula to the trace (Eq. 3) and determinant (Eq. 4) constraints ten polynomial equations are given. They can be formed as $\mathbf{Q}\mathbf{y} = \mathbf{b}$, where $\mathbf{Q}$ and $\mathbf{b}$ are the coefficient matrix and the inhomogeneous part (coefficients of monomial 1), respectively. Vector $\mathbf{y} = \begin{bmatrix} \alpha^3 & \beta^3 & \alpha^2\beta & \alpha\beta^2 & \alpha^2 & \beta^2 & \alpha\beta & \alpha & \beta \end{bmatrix}$ consists of the monomials of the system. $\mathbf{Q}$ is of size 10×9, therefore, the system is solvable and overdetermined since ten equations are given for nine unknowns. Its optimal solution in least squares sense is given by $\mathbf{y} = \mathbf{Q}^{\dagger}\mathbf{b}$, where matrix $\mathbf{Q}^{\dagger}$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{Q}$.

The elements of the solution vector $\mathbf{y}$ are dependent. Thus α and β can be obtained in multiple ways, e.g. as $\alpha_1 = y_8$, $\beta_1 = y_9$ or $\alpha_2 = \sqrt[3]{y_1}$, $\beta_2 = \sqrt[3]{y_2}$. To choose the best candidates, we paired every possible α and β, thus obtaining nine solutions, and selected the one minimizing Eq. 3, i.e. the trace constraint. The fundamental matrix is finally calculated as $\mathbf{F} = \mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-1}$.

Remark that for applications not requiring real time performance, numerically optimizing α and β to minimize Eq. 3 is a straightforward choice. Nevertheless, to our experiments, the method leads to stable results without additional optimization.

C. Transformation of Local Affinities by the Camera Matrices

The aim of this section is to show how the multiplication of the point coordinates by the intrinsic parameters modifies the

corresponding local affinities. Unlike to the rest of the paper, we assume here that points $\mathbf{p}$ and $\mathbf{p}'$ are not multiplied by $\mathbf{K}^{-1}$ and $\mathbf{K}'^{-1}$. The original relationship between the affine parameters comes from Eq. 7 by replacing the normals with $\mathbf{F}^{\mathrm{T}}\mathbf{p}'$ and $\mathbf{F}\mathbf{p}$ as follows:

$$(\hat{\mathbf{A}}^{-\mathrm{T}}\mathbf{F}^{\mathrm{T}}\mathbf{p}')_{(1:2)} = -(\mathbf{F}\mathbf{p})_{(1:2)}, \tag{11}$$

where $\hat{\mathbf{A}}$ is of size 3×3 as follows:

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & 1 \end{bmatrix}.$$

Because of $\mathbf{F} = \mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-1}$, Eq. 11 is modified as

$$(\hat{\mathbf{A}}^{-\mathrm{T}}\mathbf{K}^{-\mathrm{T}}\mathbf{E}^{\mathrm{T}}\mathbf{K}'^{-1}\mathbf{p}')_{(1:2)} = -(\mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-1}\mathbf{p})_{(1:2)}.$$

Let us denote $\mathbf{K}'^{-1}\mathbf{p}'$ and $\mathbf{K}^{-1}\mathbf{p}$ with $\mathbf{q}'$ and $\mathbf{q}$, respectively. After elementary modifications, it can be written as

$$(\mathbf{E}^{\mathrm{T}}\mathbf{q}')_{(1:2)} = -(\mathbf{K}^{\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{q})_{(1:2)}.$$

Therefore, due to the transformation of the intrinsic parameters, the original local affinity $\mathbf{A}$ must be modified as

$$\widetilde{\mathbf{A}} = (\mathbf{K}'^{-1}\hat{\mathbf{A}}\mathbf{K})_{(1:2,1:2)}. \tag{12}$$

However, matrix $\mathbf{A}$ remains the same if $\mathbf{K} = \mathbf{K}'$ and the shear is zero for both cameras.

Note that this is a mandatory step if the two images are taken by cameras with different intrinsic parameters.

D. Normalization of Affine Parameters

It is well-known that numerical instability makes the normalization of the input data essential [19]. After normalizing the point coordinates, the measured affine transformation are not valid any more (w.r.t. the normalized coordinates), they have to be normalized as well. Let us denote the normalizing transformations in the two images by $\mathbf{T}_1$ and $\mathbf{T}_2$ which translate the point sets into the origin and their mean distance from that to $\sqrt{2}$. The normalization of the point coordinates (which have been premultiplied by the intrinsic parameters) is trivial as $\widetilde{\mathbf{p}} = \mathbf{T}_1\mathbf{p}$ and $\widetilde{\mathbf{p}}' = \mathbf{T}_2\mathbf{p}'$ [2]. The normalized essential matrix can be calculated from the original as follows: $\widetilde{\mathbf{E}} = \mathbf{T}_2^{-\mathrm{T}}\mathbf{E}\mathbf{T}_1^{-1}$. After point normalization the relationship of the essential matrix and the affine transformation (Eq. 7) is modified as follows:

$$(\hat{\mathbf{A}}^{-\mathrm{T}}(\mathbf{T}_2^{\mathrm{T}}\widetilde{\mathbf{E}}\mathbf{T}_1)^{\mathrm{T}}\mathbf{p}')_{(1:2)} = -(\mathbf{T}_2^{\mathrm{T}}\widetilde{\mathbf{E}}\mathbf{T}_1\mathbf{p})_{(1:2)},$$

where $\hat{\mathbf{A}}$ is the same 3×3 matrix as in the previous section. After elementary modifications, it can be written as

$$(\widetilde{\mathbf{E}}^{\mathrm{T}}\mathbf{T}_2\mathbf{p}')_{(1:2)} = -(\mathbf{T}_1^{-\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{T}_2^{\mathrm{T}}\widetilde{\mathbf{E}}\mathbf{T}_1\mathbf{p})_{(1:2)}.$$

Thus

$$\widetilde{\mathbf{A}}^{\mathrm{T}} = (\mathbf{T}_1^{-\mathrm{T}}\hat{\mathbf{A}}^{\mathrm{T}}\mathbf{T}_2^{\mathrm{T}})_{(1:2,1:2)}.$$

The normalized affine transformation $\widetilde{\mathbf{A}}$ is calculated as

$$\widetilde{\mathbf{A}} = (\mathbf{T}_2\hat{\mathbf{A}}\mathbf{T}_1^{-1})_{(1:2,1:2)}.$$

Note that this equation is the same as Eq. 12 and holds for all transformations that can be written by 3×3 matrices e.g. the camera intrinsic parameters and the normalizing transformations in the image space.

The affinities used during the estimation are normalized by both the normalizing transformations and the intrinsic parameters. Thus affine transformation $\mathbf{A}$ is modified as follows:

$$\overline{\mathbf{A}} = (\mathbf{T}_2\mathbf{K}'^{-1}\hat{\mathbf{A}}\mathbf{K}\mathbf{T}_1^{-1})_{(1:2,1:2)}$$

Note that the proposed normalization is possible only if more than two correspondences are given. Otherwise, only the normalization by the intrinsic parameters is required.

IV. Experimental Results

The proposed method is validated both on synthesized and real world data in this section. A Matlab implementation is included as Alg. 1.[1]

A. Validation on Synthesized Tests

In order to test the proposed method in a fully controlled synthetic environment, two perspective cameras are generated by their projection matrices $\mathbf{P}$ and $\mathbf{P}'$. Their common intrinsic parameters are focal lengths $f_x = f_y = 600$ and principal point $\mathbf{p}_0 = \begin{bmatrix}300 & 300\end{bmatrix}^{\mathrm{T}}$. For the tests, three types of camera motions are considered: forward, sideways and random motions. The lengths of these motions are 2 and the distances of the plane origins from the camera centers are 10 along axis Z and around 0.1 along axes X and Y. We do not check whether a point is visible on both cameras or not since it does not affect the results of the methods. Having more than one plane is required to get a non-degenerate set up, thus points are sampled on 100 different random planes and projected onto the cameras. Zero-mean Gaussian-noise is added to the point locations. Homography is calculated using the plane parameters [2]. The affine transformation related to each point pair is calculated exploiting the noisy coordinates and the ground truth homography as it is given in [22]:

$$a_1 = \frac{h_{11} - h_{31}u'}{s} \quad a_2 = \frac{h_{21} - h_{31}v'}{s}$$
$$a_3 = \frac{h_{12} - h_{32}u'}{s} \quad a_4 = \frac{h_{22} - h_{32}v'}{s}$$

where h_{ij} $(i, j \in \{1, 2, 3\})$ is an element of the homography matrix, $s = \mathbf{h}_3^{\mathrm{T}}[u \quad v \quad 1]^T$ and $\mathbf{h}_3^{\mathrm{T}}$ is the last row the homography. The obtained essential matrices are decomposed into translation and rotation components [2] and compared to the ground truth motion.

The error of an estimated rotation matrix was calculated as follows:

$$e_{\mathrm{r}} = |\ \mathrm{rodrigues}(\mathbf{R}_{\mathrm{gt}}^{\mathrm{T}}\mathbf{R}_{\mathrm{est}})\ | \tag{13}$$

where $\mathbf{R}_{\mathrm{gt}}$ is the ground truth and $\mathbf{R}_{\mathrm{est}}$ is the estimated rotation matrices. Function rodrigues converts a rotation matrix to vector $\mathbf{r} \in \mathbb{R}^3$ where $|\mathbf{r}|$ is the angle of rotation around axis $\mathbf{r}/|\mathbf{r}|$. Since the length of the translation vector cannot be recovered due to the scale ambiguity of the perspective projection, the error of the translation vector is the angle (in

[1] C++ implementation is available at http://web.eee.sztaki.hu/~dbarath/.

degrees) between the estimated and ground truth vectors. It is as follows:

$$e_{\mathrm{t}} = \mathrm{acos}(\mathbf{t}_{\mathrm{gt}}^{\mathrm{T}}\mathbf{t}_{\mathrm{est}}), \quad |\mathbf{t}_{\mathrm{gt}}| = |\mathbf{t}_{\mathrm{est}}| = 1, \tag{14}$$

where $\mathbf{t}_{\mathrm{gt}}$ and $\mathbf{t}_{\mathrm{est}}$ are the ground truth and estimated translations, respectively.

In Fig. 3, we compare four methods: the proposed algorithm applied to two correspondences (Proposed), the normalized version of the proposed method applied to five point pairs (Normalized Prop.), the five-point algorithm [8] (Nistér) and the technique proposed in [18] (Raposo et al.). The top row shows the mean error (vertical axis) of the obtained rotation matrices plotted as the function of the noise σ (horizontal axis). The bottom row reports the quality of the estimated translation vectors. The mean angular error (in radians, vertical axis) w.r.t. the ground truth translation is plotted as the function of the noise σ (horizontal axis).

For the **first column** of Fig. 3, forward motion and no rotation is applied to the cameras. It can be seen that the proposed method exploiting two correspondences outperforms both the five-point algorithm and that of Raposo et al. The translation vector obtained by the normalized algorithm is sensitive to this kind of motion, however, the estimated rotation matrix is the most accurate. The **second column** reports the error if only sideways motion is considered. In these tests, the proposed method and that of Raposo et al. achieved similar accuracy. The normalized version is superior to all competitor methods in both terms. If random motion is applied (**third column**), the rotation obtained by the proposed two-point algorithm outperforms both the methods of Nistér and Raposo et al. while achieving similar results to Raposo et al. for the translation vector. The normalized algorithm provided the most accurate results in both aspects. **The last column** reports the results for nearly planar scenes. Only a small Gaussian-noise with 10^{-5} standard deviation is added to the plane tangents having the same base point. It can be seen that the 5-point algorithm leads to the most accurate translation vectors, however, the proposed methods outperform the competitor ones for estimating the camera rotation.

Concluding the synthesized tests, the proposed algorithm (without normalization) outperforms the competitor ones in four out of the eight tests and achieve similar results in the remaining ones. The normalized version applied to five correspondences is superior to all methods in both terms except two test cases.

B. Real World Experiments

To test the proposed solver on real world images, we downloaded the `strecha` dataset [23] consisting of image sequences of buildings. All images are of size 3072×2048. The ground truth projection matrices are provided. The methods were applied to all possible image pairs in each sequence. The Hessian-Affine detector [11] encapsulated into the view-synthesizer of ASIFT [12] was used to obtain affine covariant correspondences. This combination performed the best in [24]. For each image pair, a reference point set with ground truth inliers was obtained by calculating the fundamental matrix from the projection matrices [2]. Correspondences were considered as inliers if the symmetric epipolar distance was smaller than 1.0 pixel. All image pairs with less than 50 inliers were discarded. Also, pairs were removed where none of the methods found the ground truth essential matrix. In total, 714 pairs were used in the evaluation. The used errors of the rotations and translations were the same as the ones which were used for the synthesized tests.

As a robust estimator, we chose Graph-Cut RANSAC [25] since it can be considered as state-of-the-art variant of RANSAC, and its implementation is publicly available[2]. The scoring function, i.e. the one determining the quality of a model, was set to the MSAC-like truncated quadratic cost [26] with noise σ set to 0.3 pixels (proposed in [27]). The point-to-model residual function was the Sampson-distance. To estimate essential matrices from a non-minimal sample, we chose the normalized eight-point algorithm [19]. The minimum iteration number was set to 100. Other parameters were set to the default values of [25]. Note that instead of the eight-point algorithm, the proposed normalized method could also be used. However, to our experiments, the proportion of the outliers in the set of affine transformations is often high. Thus the least-squares fitting can fail.

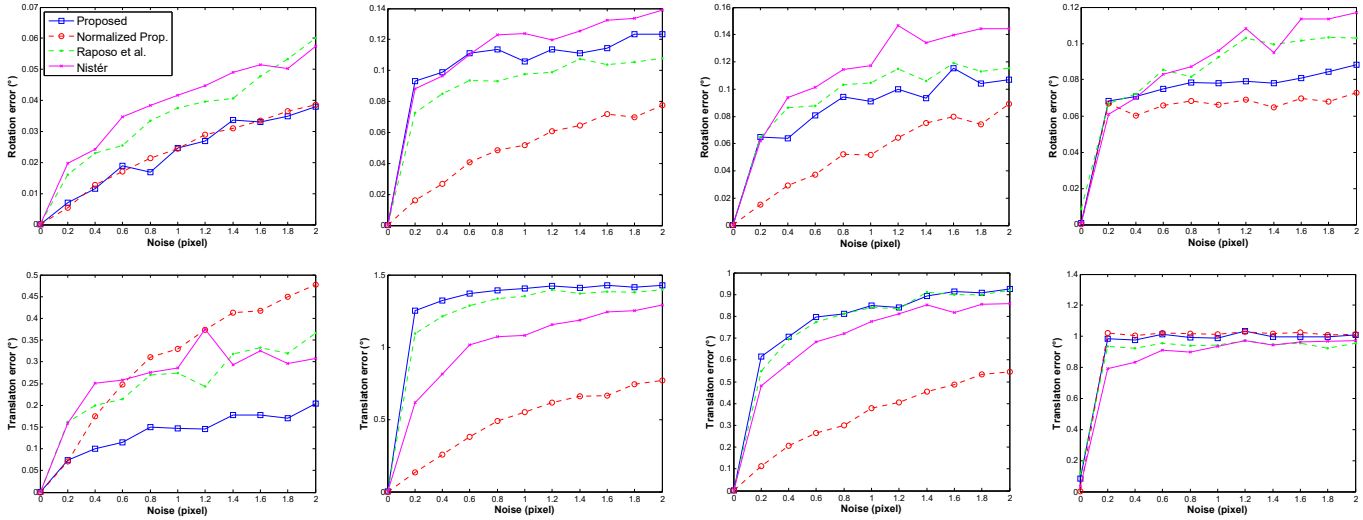
Table I reports the results of GC-RANSAC combined with minimal methods. The competitor ones were the proposed 2-point algorithm, 3-point[3] [17], 5-point[4] [8], and the method of Raposo et al. [18] techniques. The first two columns show the sequence and the number of image pairs (Pair #). The mean errors of the translation (e_{t}; in degrees) and rotation (e_{r}; in degrees) are shown in the first two columns regarding to each minimal method. Even though the differences are fairly small, i.e. under a degree, the proposed solver leads to the most accurate results with four times less iterations than what the five-point algorithm requires. Fig. 4 contains example results of the proposed method with inliers (circles) and outliers (black crosses) drawn.

C. Processing Time

The proposed algorithm consists of two main steps. First, the null space of a 6×9 matrix is calculated. Then the final solution is given as the pseudo-inverse of a matrix of size 10×9. Both steps have negligible time demand, therefore, the proposed algorithm is applicable even to online tasks. The generalization to n correspondences modifies only the first matrix to size $3n \times 9$ ($n \geq 2$). The mean processing time of 1000 runs of the 2-point version implemented in C++ is approx. 530μsecs (53×10^{-5} seconds). The time demand of the n-point version, i.e. the overdetermined case, is around 49 msecs (49×10^{-3} seconds) for $n = 4000$.

Augmenting a robust estimator, e.g. RANSAC [29], with the 2-point algorithm is beneficial since it yields significantly faster convergence. See Table II reporting the theoretical iteration number of RANSAC combined with different minimal

[2]https://github.com/danini/graph-cut-ransac
[3]Own implementation is used.
[4]Available at http://nghiaho.com/?p=1675



(a) Forward motion (b) Sideways motion (c) Random motion (d) Nearly planar scene

Fig. 3. The errors (vertical axes) of the estimated rotations (top row; in radian; Eq. 13) and translations (bottom; in radian; Eq. 14) plotted as the function of the noise σ (horizontal axes; in pixels). Each column represents a camera motion: (a) pure forward, and (b) sideways motion, (c) random motion, and (d) nearly planar scene with cameras having random motion. The errors are the mean of 1000 runs on each noise σ. The reported algorithms: the proposed one applied to a minimal sample (Proposed), the normalized version of the proposed method applied to five correspondences (Normalized Prop.), the technique of Raposo and Barreto [18], and the 5-point algorithm proposed by David Nister [8].

TABLE I

ACCURACY OF MINIMAL METHODS FOR RELATIVE MOTION ESTIMATION ON THE STRECHA DATASET [28] (6 SEQUENCES AND THUS 714 IMAGE PAIRS). GC-RANSAC [25] WAS USED AS ROBUST ESTIMATOR. THE FIRST TWO COLUMNS SHOW THE SEQUENCES AND THE NUMBERS OF IMAGE PAIRS (PAIR #). OTHER COLUMNS REPORT THE AVERAGE RESULTS (10 RUNS ON EACH IMAGE PAIR) OF THE COMPETITOR METHODS AT 95% CONFIDENCE. THE MEAN ERROR OF THE OBTAINED TRANSLATIONS (e_T; IN DEGREES) AND ROTATIONS (e_R; IN DEGREES), THEIR STANDARD DEVIATION, AND THE NUMBER OF REQUIRED ITERATIONS FOR GC-RANSAC (s) ARE WRITTEN INTO THE THREE COLUMNS REGARDING TO EACH METHOD. SEQUENCES: (A) FOUNTAIN-P11, (B) ENTRY-P10, (C) HERZJESUS-P8, (D) CASTLE-P19, (E) CASTLE-P30, (F) HERZJESUS-P25. EXAMPLE IMAGE PAIRS ARE IN FIGURE 4.

		Nistér et al. [8]			Raposo et al. [18]			Bentolila et al. [17]			**Proposed**		
	Pair #	e_t	e_r	s	e_t	e_r	s	e_t	e_r	s	e_t	e_r	s
(a)	55	0.17 ± 0.14	0.35 ± 0.51	200	0.20 ± 0.51	0.37 ± 0.51	**100**	0.24 ± 0.53	0.37 ± 0.60	132	**0.15** ± 0.12	**0.34** ± 0.51	**100**
(b)	17	0.27 ± 0.29	0.29 ± 0.33	114	0.25 ± 0.33	**0.27** ± 0.30	**100**	0.21 ± 0.14	0.28 ± 0.34	142	0.35 ± 0.42	0.28 ± 0.33	**100**
(c)	28	0.18 ± 0.14	0.16 ± 0.09	110	0.26 ± 0.09	0.15 ± 0.07	**100**	0.19 ± 0.13	0.14 ± 0.06	172	**0.17** ± 0.14	**0.13** ± 0.05	117
(d)	88	1.15 ± 2.66	0.14 ± 0.09	502	1.29 ± 0.09	0.16 ± 0.09	**100**	0.99 ± 2.81	**0.13** ± 0.08	197	**1.03** ± 2.61	0.14 ± 0.08	102
(e)	251	1.47 ± 5.63	0.14 ± 0.08	602	1.76 ± 6.84	0.16 ± 0.09	**105**	2.09 ± 8.83	**0.13** ± 0.08	201	**1.40** ± 4.77	**0.13** ± 0.07	106
(f)	275	**0.36** ± 0.15	0.15 ± 0.09	538	0.37 ± 1.01	0.15 ± 0.07	110	0.37 ± 1.26	**0.13** ± 0.09	235	**0.36** ± 1.06	**0.13** ± 0.13	**108**
all	714	0.79 ± 3.49	0.20 ± 6.82	490	0.94 ± 4.27	0.21 ± 0.19	**105**	1.03 ± 5.38	0.20 ± 0.21	205	**0.76** ± 3.00	**0.19** ± 0.19	**105**

methods. It is clear that the estimation exploiting two correspondences is advantageous to achieve real time performance even for high outlier ratio.

TABLE II

REQUIRED THEORETICAL ITERATION NUMBER OF RANSAC AUGMENTED WITH MINIMAL METHODS (COLUMNS) WITH 95% PROBABILITY ON DIFFERENT OUTLIER LEVELS (ROWS).

	# of required points					
Outl.	**2**	3	5	6	7	8
80%	**74**	$\sim 10^3$	$\sim 10^4$	$\sim 10^5$	$\sim 10^5$	$\sim 10^6$
95%	**1 197**	$\sim 10^4$	$\sim 10^7$	$\sim 10^8$	∞	∞
99%	**29 856**	$\sim 10^6$	∞	∞	∞	∞

D. Application: Multi-motion Fitting

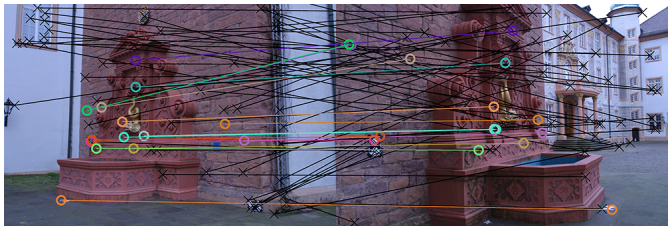
The clustering of correspondences to multiple rigid motions in two-views is usually solved by applying a multi-model fitting algorithm, e.g. PEARL [30] or Multi-X [31], combined with a minimal method as an engine estimating fundamental matrices. Recent approaches are based on a RANSAC-like initialization, therefore, their results highly depend on the applied minimal method, especially, on the size of the minimal sample – the probability of finding an accurate model increases if the model is estimable using less correspondences.

Table III reports the results of Multi-X method fitting multiple rigid motions, i.e. fundamental matrices, simultaneously. Each row contains the results of a minimal method: the seven- (7PT) and eight-point (8PT) algorithms and the proposed one (2PT). The errors are the misclassification errors (ME), i.e. the ratio of misclassified correspondences:

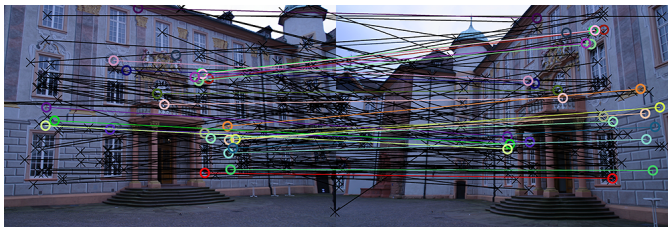
$$\text{ME} = \frac{\#\text{Misclassified Points}}{\#\text{Points}},$$

reported in percentage. Columns are the test pairs of the AdelaideRMF dataset[5] which consists of 18 image pairs of size 640×480 each containing point correspondences assigned to rigid motions manually. Since the proposed method requires

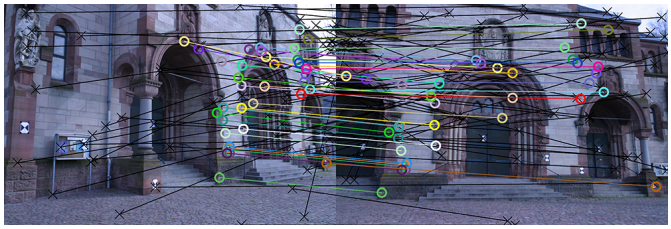
[5]https://cs.adelaide.edu.au/~hwong/doku.php?id=data



(a) `fountain-p11`



(b) `entry-p10`



(c) `herzjesus-p8`

(d) `castle-p19`

Fig. 4. Example results of the proposed algorithm on image pairs from the `strecha` dataset. Inliers drawn by circles and outliers by black crosses. Every 10th correspondence is drawn. The used robust estimator is GC-RANSAC [25]. Quantitative evaluation is in Table I.

affine correspondences, we applied AHessian-Affine to the image pairs detecting as many correspondences as we can. For all annotated correspondences, i.e. the point pairs provided in the dataset, we searched the closest match in the detected correspondence set, and replaced them with the matched ones. Note that this could introduce error into the annotation, however, these point pairs are used for all tests, including the proposed and competitor methods, thus the comparison remains fair.

Since we aim at estimating essential matrices, the intrinsic camera calibration have to be known a priori. We estimated those intrinsic parameters for each image pair from the manually annotated point correspondences by the following procedure. We assumed the semi-calibrated case: the principal point was set to the center of the image and the pixel ratio to one. It was assumed that the images in each pair have the same focal length f. In order to recover f, we applied [6] [6] to a number of 6-sized subsets (20 times the point number of the current motion) of the ground truth correspondences regarding to each motion. Finally, weighted histogram voting [32] was used to select the best candidate out of the obtained focal lengths.

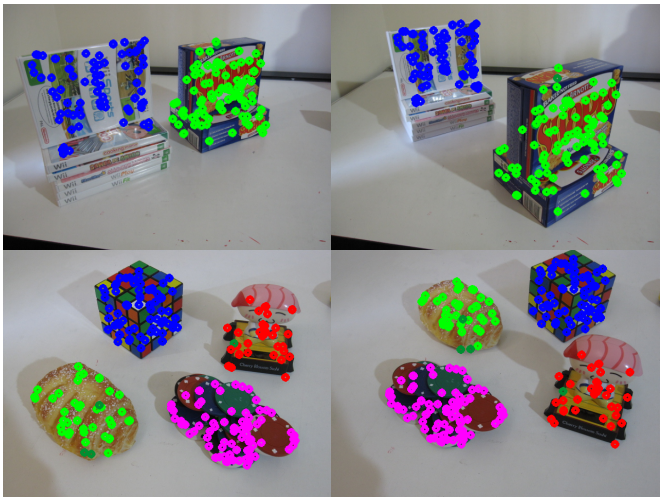


Fig. 5. Example two-view multi-motion fitting on pairs `Gamebiscuit` and `Cubebreadtoychips` from the AdelaideRMF dataset. Color denotes motions.

According to Table III, Multi-X leads to the most accurate clusterings, in terms of misclassification error, if it is combined with the proposed two-point algorithm.

V. Conclusion

It is shown in this paper that a local affine transformation yields two linear constraints for essential matrix estimation. Exploiting these constraints, the essential matrix can efficiently be recovered using two affine correspondences. Even though the proposed solution assumes perspective camera model, it can straightforwardly be generalized to arbitrary one, e.g. omni-directional cameras. Also, the normalization of the affine parameters are shown that is mandatory if the intrinsic camera parameters differ or the point coordinates are normalized. It is validated both on synthesized tests and 714 real image pairs that combining the proposed solver with recent robust estimators, e.g. Graph-Cut RANSAC, leads to results superior to the state-of-the-art both in terms of geometric accuracy and number of samples required.

Acknowledgement

This research was supported by the Hungarian Scientific Research Fund (No. `OTKA/NKFIH 120499`) and the European Union, co-financed by the European Social Fund (`EFOP-3.6.3-VEKOP-16-2017-00001`). Daniel Barath acknowledges the support of the OP VVV funded project `CZ.02.1.01/0.0/0.0/16_019/0000765` "Research Center for Informatics".

[6] http://cmp.felk.cvut.cz/mini/

TABLE III
TWO-VIEW MULTI-MOTION FITTING ON THE ADELAIDERMF DATASET USING MULTI-X METHOD AUGMENTED WITH DIFFERENT MINIMAL METHODS (ROWS): THE PROPOSED TWO-POINT ALGORITHM (2PT), THE SEVEN-POINT (7PT) AND EIGHT-POINT (8PT) METHODS. THE REPORTED ERRORS ARE MISCLASSIFICATION ERRORS IN PERCENTAGE, I.E. THE RATIO OF THE MISCLASSIFIED CORRESPONDENCES. TEST PAIRS: (1) BISCUITBOOKBOX, (2) BREADCARTOYCHIPS, (3) BREADCUBECHIPS, (4) BREADTOYCAR, (5) CARCHIPSCUBE, (6) CUBEBREADTOYCHIPS, (7) DINOBOOKS, (8) TOYCUBECAR, (9) BISCUIT, (10) BOARDGAME, (11) BOOK, (12) BREADCUBE, (13) BREADTOY, (14) CUBE, (15) CUBETOY, (16) GAME, (17) GAMEBISCUIT, (18) CUBECHIPS.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	AVG	MED
2PT	5.0	**5.1**	2.2	**7.2**	**6.1**	4.9	**7.2**	**5.5**	**29.4**	**8.2**	**2.7**	5.2	11.5	**27.8**	**3.7**	**7.3**	3.7	**7.0**	**8.3**	**5.8**
7PT	**3.9**	5.5	**1.7**	7.8	**6.1**	**4.3**	11.4	6.0	30.3	8.6	**2.7**	**2.5**	11.8	29.8	5.2	7.7	3.0	8.1	8.7	6.1
8PT	4.6	8.4	2.2	**7.2**	7.3	6.1	10.6	6.5	32.1	8.6	**2.7**	3.3	**8.3**	28.5	4.8	8.6	**2.7**	9.2	9.0	7.3

APPENDIX A
PROOF OF THE LINEAR AFFINE CONSTRAINTS

It is trivial that an affine transformation $\mathbf{A}$ transforms the direction of the corresponding epipolar lines to each other as all affine transformations correctly modify the lines going through the corresponding point locations $[u \quad v]$ and $[u' \quad v']$. Therefore, $\mathbf{A}\mathbf{v} \parallel \mathbf{v}'$, where $\mathbf{v}$ and $\mathbf{v}'$ are the directions of the epipolar lines on the first and second images.

As it is well-known in computer graphics [21], line normals are transformed as $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \beta\mathbf{n}'$, where $\mathbf{n} = (\mathbf{F}^{\mathrm{T}}\mathbf{p}')_{1:2}$ and $\mathbf{n}' = (\mathbf{F}\mathbf{p})_{1:2}$ are the normals of the epipolar lines ($\beta \neq 0$). Lower index $(1:2)$ denotes the first two elements of a vector. We prove here that

$$\mathbf{A}^{-\mathrm{T}}\mathbf{n} = -\mathbf{n}'. \tag{15}$$

Suppose that corresponding point pair $\mathbf{p} = [u \quad v \quad 1]^{\mathrm{T}}$ and $\mathbf{p}' = [u' \quad v' \quad 1]^{\mathrm{T}}$ are given. Let $\mathbf{n} = [n_u \quad n_v]^{\mathrm{T}}$ and $\mathbf{n}' = [n'_u \quad n'_v]^{\mathrm{T}}$ be the normal directions of epipolar lines

$$\mathbf{l}_1 = \mathbf{F}^{\mathrm{T}}\mathbf{p}' = [l_{1,a} \quad l_{1,b} \quad l_{1,c}]^{\mathrm{T}}, \tag{16}$$

and

$$\mathbf{l}'_1 = \mathbf{F}\mathbf{p} = [l'_{1,a} \quad l'_{1,b} \quad l'_{1,c}]^{\mathrm{T}}, \tag{17}$$

respectively. It is trivial that $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \beta\mathbf{n}'$ due to $\mathbf{A}\mathbf{v} \parallel \mathbf{v}'$, where β is a scale factor. First, it is shown how affine transformation $\mathbf{A}$ transforms the length of $\mathbf{n}$ if it is a unit vector. To calculate this scale factor β, it is required to introduce a new point as close to $\mathbf{p}$ as possible determining epipolar lines on both images and β as the ratio of distances from these new lines. Let us introduce point $\mathbf{q} = \mathbf{p} + \delta\begin{bmatrix}\mathbf{n}^T & 0\end{bmatrix}^T$, where δ is a small scalar value. Point $\mathbf{q}$ determines an epipolar line $\mathbf{l}'_2 = [l'_{2,a} \quad l'_{2,b} \quad l'_{2,c}]^{\mathrm{T}}$ on the second image as

$$\mathbf{l}'_2 = \mathbf{F}\mathbf{q} = \mathbf{F}\left(\mathbf{p} + \delta\begin{bmatrix}\mathbf{n}^{\mathrm{T}} & 0\end{bmatrix}^{\mathrm{T}}\right) = [s_1 \quad s_2 \quad s_3]^{\mathrm{T}},$$

where

$$\begin{aligned} s_1 &= l'_{1,a} + \delta f_{11}n_u + \delta f_{12}n_v, \\ s_2 &= l'_{1,b} + \delta f_{21}n_u + \delta f_{22}n_v, \\ s_3 &= l'_{1,c} + \delta f_{31}n_u + \delta f_{32}n_v. \end{aligned}$$

Then scale β is given by the distance d' between line $\mathbf{l}'_2$ and point $\mathbf{p}'$. The setup is visualized in Fig. 1(b). The calculation of distance d' is given by the well-known formula as follows:

$$d' = \frac{|s_1u' + s_2v' + s_3|}{\sqrt{s_1^2 + s_2^2}}. \tag{18}$$

It is known that point $\mathbf{p}'$ lies on $\mathbf{l}'_1$, which can be written as $l'_{1,a}u' + l'_{1,b}v' + l'_{1,c} = 0$. This fact reduces Eq. 18 to

$$\begin{aligned} d' &= \frac{|\hat{s}_1u' + \hat{s}_2v' + \hat{s}_3|}{\sqrt{s_1^2 + s_2^2}}, \\ \hat{s}_1 &= \delta f_{11}n_u + \delta f_{12}n_v, \\ \hat{s}_2 &= \delta f_{21}n_u + \delta f_{22}n_v, \\ \hat{s}_3 &= \delta f_{31}n_u + \delta f_{32}n_v. \end{aligned} \tag{19}$$

To determine β, the introduced point $\mathbf{q}$ has to be moved infinitesimally close to the location of $\mathbf{p}$. In other words, $\delta \to 0$. β is the ratio of the length of vector $(\mathbf{p} - \mathbf{q})$ and the distance between point $\mathbf{p}'$ and line $\mathbf{l}'_2$. The latter is δ, while the former has just calculated in Eq. 19. Therefore the square of β is written as

$$\beta^2 = \lim_{\delta\to 0}\frac{\delta^2}{d'^2} = \lim_{\delta\to 0}\frac{\delta^2\left(s_1^2 + s_2^2\right)}{|\hat{s}_1u' + \hat{s}_2v' + \hat{s}_3|^2}. \tag{20}$$

After elementary modifications, the final formula for scale β is given as

$$\begin{aligned} \beta &= \pm\frac{\sqrt{l'_{1,a}l'_{1,a} + l'_{1,b}l'_{1,b}}}{|\widetilde{s}_1u' + \widetilde{s}_2v' + \widetilde{s}_3|}, \\ \widetilde{s}_i &= f_{i1}n_u + f_{i2}n_v, \quad i \in \{1, 2, 3\}. \end{aligned} \tag{21}$$

The epipolar line corresponding to point $\mathbf{p}$ is parameterized as $\begin{bmatrix}l'_{1,a} & l'_{1,b} & l'_{1,c}\end{bmatrix} = \mathbf{F}[u \quad v \quad 1]^{\mathrm{T}}$. Therefore, the normal of the line is as $\mathbf{n}' = \begin{bmatrix}l'_{1,a} & l'_{1,b}\end{bmatrix}^{\mathrm{T}} = (\mathbf{F}\begin{bmatrix}u' & v' & 1\end{bmatrix}^{\mathrm{T}})_{(1:2)}$. Similarly, $\mathbf{n} = (\mathbf{F}^{\mathrm{T}}\begin{bmatrix}u' & v' & 1\end{bmatrix}^{\mathrm{T}})_{(1:2)}$. The numerator in Eq. 21 can be rewritten as $|\mathbf{n}| = \sqrt{l_{1,a}^2 + l_{1,b}^2}$, while the denominator is as follows:

$$\begin{aligned} \widetilde{s}_1u' + \widetilde{s}_2v' + \widetilde{s}_3 = n_u(f_{11}u' + f_{21}v' + f_{31}) + \\ n_v(f_{12}u' + f_{22}v' + f_{32}) = n_u^2 + n_v^2 = |\mathbf{n}|^2. \end{aligned}$$

Thus

$$\beta = \pm\frac{|\mathbf{n}|}{|\mathbf{n}|^2} = \pm\frac{1}{|\mathbf{n}|}.$$

The length of normal $\mathbf{n}$ is one, thus $\beta = 1$, and Eq. 5 is modified as $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = \pm\mathbf{n}'$. Since the direction of the epipolar lines on the two images must be the opposite of each other, the positive solution can be omitted. The final formula is as follows: $\mathbf{A}^{-\mathrm{T}}\mathbf{n} = -\mathbf{n}'$.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TIP.2018.2849866

Program 1: The Two-point Algorithm

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 2-pt algorithm.
%% Use Matlab-7.0(6.5) with SymbolicMath Toolbox.
%% Input:
%%      The "Matches" is a 2x8 matrix containing two affine correspondences.
%%      Each row of "Matches": (u1, v1, u2, v2, a1, a2, a3, a4).
%%      "K1" and "K2" are two calibration matrices.
%% Output: fundamental matrix.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function F = TwoPointFundamental(Matches, K1, K2)
        syms E e x y equ C
        equ = sym('equ', [1 10]);
        C = sym('C', [10 10]);

        M = zeros(6, 9)
        for i = 1 : 2
                u1 = Matches(i, 1); v1 = Matches(i, 2); u2 = Matches(i, 3); v2 = Matches(i, 4);
                a1 = Matches(i, 5); a2 = Matches(i, 6); a3 = Matches(i, 7); a4 = Matches(i, 8);

                M(3*(i-1) + 1 : 3*i, :) = ...
                        [u1 * u2, v1 * u2, u2, u1 * v2, v1 * v2, v2, u1, v1, 1;
                        u2 + a1 * u1, a1 * v1, a1, v2 + a3 * u1, a3 * v1, a3, 1, 0, 0;
                        a2 * u1, u2 + a2 * v1, a2, a4 * u1, v2 + a4 * v1, a4, 0, 1, 0];
        end

        N = null(M); %%% Compute the null-space
        e = x*N(:,1) + y*N(:,2) + N(:,3);
        E = transpose(reshape(e,3,3));
        ET = transpose(E);

        equ(1) = det(E);
        equ(2:10) = expand(2*E*ET*E-sum(diag(E*ET))*E);

        for i = 1 : 10
                equ(i) = maple('sort', maple('collect', equ(i), '[x,y]', 'distributed'));
                for j = 1 : 9
                        oper = maple('op', j, equ(i));
                        C(i,j) = maple('op', 1, oper);
                end
                C(i,10) = maple('op', 10, equ(i));
        end

        nC = double(C); %%% Convert the coefficient matrix to numeric format
        Res = pinv(nC(:,1:9)) * (-nC(:,10)); %%% Compute alpha and beta
        alpha = Res(8); beta = Res(9);

        nE = alpha*N(:,1) + beta*N(:,2) + N(:,3); %%% Compute the essential matrix
        nE = transpose(reshape(Essential,3,3));

        F = inv(K2') * nE * inv(K1) %%% Get the fundamental matrix
end
```

References

[1] Q.-T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *International Journal of Computer Vision*, 1996.

[2] R. I. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[3] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky, "A minimal solution for relative pose with unknown focal length," *Image and Vision Computing*, 2008.

[4] H. Li, "A simple solution to the six-point two-view focal-length problem," in *European Conference on Computer Vision*. Springer, 2006.

[5] W. Wang and C. Wu, "Six-point synthetic method to estimate fundamental matrix," *Science in China Series E: Technological Sciences*, 1997.

[6] R. I. Hartley and H. Li, "An efficient hidden variable approach to minimal-case camera motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[7] J. Philip, "A non-iterative algorithm for determining all essential matrices corresponding to five point pairs," *The Photogrammetric Record*, 1996.

[8] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[9] H. Li and R. I. Hartley, "Five-point motion estimation made easy," in *International Conference on Pattern Recognition*. IEEE, 2006.

[10] D. Batra, B. Nabbe, and M. Hebert, "An alternative formulation for five point relative pose problem," in *IEEE Workshop on Motion and Video Computing*. IEEE, 2007.

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, 2005.

[12] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, 2009.

[13] D. Mishkin, J. Matas, and M. Perdoch, "MODS: Fast and robust method for two-view matching," *Computer Vision and Image Understanding*, 2015.

[14] M. Perd'och, J. Matas, and O. Chum, "Epipolar geometry from two correspondences," in *18th International Conference on Pattern Recognition*. IEEE.

[15] O. Chum, J. Matas, and S. Obdrzálek, "Epipolar geometry from three correspondences," *Computer Vision Winter Workshop*, 2003.

[16] D. Barath, J. Molnar, and L. Hajder, "Novel methods for estimating surface normals from affine transformations," in *Computer Vision, Imaging and Computer Graphics Theory and Applications*. Springer International Publishing, 2016.
[17] J. Bentolila and J. M. Francos, "Conic epipolar constraints from affine correspondences," *Computer Vision and Image Understanding*, 2014.
[18] C. Raposo and J. P. Barreto, "Theory and practice of structure-from-motion using affine correspondences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5470–5478.
[19] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.
[20] D. Barath, J. Molnar, and L. Hajder, "Novel methods for estimating surface normals from affine transformations," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics*. Springer International Publishing, 2015, pp. 316–337.
[21] K. Turkowski, "Transformations of surface normal vectors," in *Tech. Rep. 22, Apple Computer*, 1990.
[22] D. Barath and L. Hajder, "Novel ways to estimate homography from local affine transformations," in *11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016.
[23] C. Strecha, R. Fransens, and L. Van Gool, "Wide-baseline stereo from multiple views: a probabilistic account," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2004.
[24] D. Barath, L. Hajder, and J. Matas, "Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry," in *27th British Machine Vision Conference*, 2016.
[25] D. Barath and J. Matas, "Graph-Cut RANSAC," *Conference on Computer Vision and Pattern Recognition*, 2018.
[26] P. H. S. Torr, "Bayesian model estimation and selection for epipolar geometry and generic manifold fitting," *International Journal of Computer Vision*, vol. 50, no. 1, pp. 35–61, 2002.
[27] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized RANSAC," in *British Machine Vision Conference*. Citeseer, 2012.
[28] 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 *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.
[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.
[30] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International journal of computer vision*, 2012.
[31] D. Barath and J. Matas, "Multi-class model fitting by energy minimization and mode-seeking," *arXiv preprint arXiv:1706.00827*, 2017.
[32] M. Bujnak, Z. Kukelova, and T. Pajdla, "Robust focal length estimation by voting in multi-view scene reconstruction," in *Asian Conference on Computer Vision*. Springer, 2009, pp. 13–24.

Daniel Barath was born in 1989 in Budapest. He is currently a Ph.D. student at the Eötvös Loránd Science University and a member of the Machine Perception Research Laboratory at the Hungarian Academy of Sciences (MTA SZTAKI). His research interests are minimal methods in computer vision and robust model estimation.

Levente Hajder was born in 1975 in Budapest. He studied computer science at Kand Klmn Polytechnics, and electrical engineering at Budapest University of Technology and Economics. His Ph.D. was received in 2008. Currently, he is a member of Machine Perception Research Laboratory at the Computer and Automation Research Institute (MTA SZTAKI) Budapest, Hungary. His research interests are 3D Computer Vision.