

# Approximate Epipolar Geometry from Six Rotation Invariant Correspondences

Dániel Baráth

MTA SZTAKI, Kende u. 13-17, Budapest H-1111, Hungary

ELTE IK, Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary

Keywords: Fundamental Matrix, Epipolar Geometry, Rotation Invariant Features, Approximation.

Abstract: We propose a method for estimating an approximate fundamental matrix from six rotation invariant feature correspondences exploiting their rotation components, e.g. provided by SIFT or ORB detectors. The cameras are not calibrated. First, a linear sub-space is calculated from the point coordinates, then the rotations are used assuming orthographic projection. It is demonstrated that combining the proposed method with Graph-cut RANSAC makes it superior to the state-of-the-art in terms of accuracy for tasks requiring a strict time limit. These tasks are practically the ones which need to be done real time. We tested the method on 203 publicly available real image pairs.

## 1 INTRODUCTION

In this paper, we aim to approximate the epipolar geometry between two *non-calibrated* cameras exploiting six rotation invariant feature correspondences in general position (see Fig. 1). The approximated fundamental matrix  $\mathbf{F} \in \mathbb{R}^{3 \times 3}$  is then used in a recent variant of locally optimized RANSAC (Chum et al., 2003) making it faster than the state-of-the-art due to the reduced number of the required points. This speedup is beneficial in online applications, i.e. when real time processing is needed, and leads to results superior to the state-of-the-art in terms of accuracy.

The common techniques to estimate the fundamental matrix when no camera parameters are known, i.e. the non-calibrated case, are the eight- and seven-point methods (Hartley and Zisserman, 2003). Both of them are widely-used in computer vision applications and have thousands of citations year-by-year. The eight-point algorithm is based on estimating the direct linear transformation which the epipolar constraint induces. The method is fast and the stability issues had already been solved by the normalization technique of Hartley (Hartley, 1997) making the technique accurate despite the noise. The seven-point algorithm enforces the rank-two constraint, i.e. the determinant of  $\mathbf{F}$  must be zero, by solving the cubic polynomial equation which it implies.

Getting more information using exclusively point correspondences is not possible. Nevertheless, se-

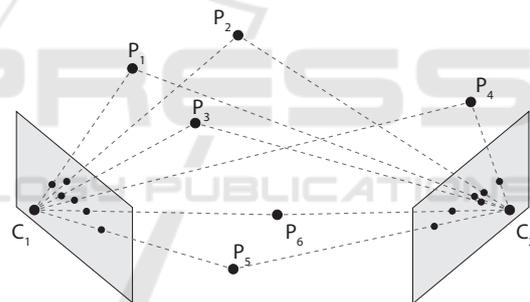


Figure 1: The projections on cameras  $C_1$  and  $C_2$  of six 3D points  $P_i$  ( $i \in [1, 6]$ ) in general position.

veral approaches had been proposed to reduce the number of unknowns. As an example, knowing the intrinsic parameters of the cameras (focal length, pixel ratio or the principal point) makes the so-called trace constraint applicable. The problem becomes solvable using six (Li, 2006; Kukulova et al., 2008; Stewénius et al., 2008; Torii et al., 2010) correspondences in the semi-calibrated case – when all intrinsic parameters are known but a common focal length. For fully calibrated cameras, five (Nistér, 2004; Li and Hartley, 2006; Batra et al., ; Kukulova et al., 2008; Hartley and Li, 2012) point pairs are enough for estimating the relative motion. One can also restrict the camera movement, e.g. the one point method proposed by Davide Scaramuzza (Scaramuzza, 2011) assumes the cameras to move on a plane and the so-called non-holonomic constraint to hold.

Approaching the problem from a different direction, it is very rare nowadays to have solely the point coordinates obtained by the feature detector. For example, SIFT (Lowe, 1999) features contain a rotation and a scale or ORB (Rublee et al., 2011) features provide a rotation. This plus information is usually not used in recent geometric model estimators. It is just thrown away at the very beginning. In this paper, we involve an additional *affine parameter*, i.e. rotation of the feature, into the fundamental matrix estimation process to reduce the size of the minimal sample required for fundamental matrix estimation.

Of course, using full affine correspondences (point pair together with rotation, shear and scales along the image axes) for epipolar geometry estimation is not a new idea. Perdoch et al. (Perdoch et al., 2006) proposed techniques for approximating the essential and fundamental matrix exploiting two and three correspondences, respectively. Barath et al. (Barath et al., 2017) showed that using two affine correspondences, the fundamental matrix and a common focal length can be estimated simultaneously. Raposo et al. (Raposo and Barreto, 2016) proposed a solution for essential matrix estimation using two correspondences. Bentolila et al. (Bentolila and Francos, 2014) proposed a method to estimate the  $\mathbf{F}$  from three correspondences proposing conic constraints.

Exploiting only a part of an affine correspondence, e.g. exclusively the rotation component, is a well-known technique for example in wide-baseline feature matching (Matas et al., 2004). To the best of our knowledge, the sole work involving them into geometric model estimation is (Barath, 2017). It assumes that  $\mathbf{F}$  is known a priori and a technique is proposed for estimating a homography using two SIFT correspondences using their scale and rotation components. Nevertheless, they consider that the scales along axes  $u$  and  $v$  equal to that of the SIFT features – which is practically not true. Thus the method obtains an approximation of the homography.

The contributions of this paper: (i) the relationship of affine correspondences and epipolar geometry proposed in (Barath et al., 2017) are reformulated making it separable to *orientation* and *scale* components. (ii) Using the proposed formulas, we assume that the orientation constraint can be satisfied by the rotation of the feature, thus making the fundamental matrix estimable from six correspondences. It is demonstrated on 203 real image pairs that the proposed method combined with a recent variant of locally optimized RANSAC outperforms the state-of-the-art if real time performance is required.

## 2 THEORETICAL BACKGROUND

**Affine Correspondences.** In this paper, we consider an affine correspondence (AC) as a triplet:  $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{A})$ , where  $\mathbf{p}_1 = [u_1 \ v_1 \ 1]^T$  and  $\mathbf{p}_2 = [u_2 \ v_2 \ 1]^T$  are a corresponding homogeneous point pair in the two images, and

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

is a  $2 \times 2$  linear transformation which we call *local affine transformation*. To define  $\mathbf{A}$ , we use the definition provided in (Molnár and Chetverikov, 2014) as it is given as the first-order Taylor-approximation of the  $3D \rightarrow 2D$  projection functions. Note that, for perspective cameras,  $\mathbf{A}$  is the first-order approximation of the related *homography* matrix

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

as follows:

$$\begin{aligned} a_1 &= \frac{\partial u_2}{\partial u_1} = \frac{h_1 - h_7 u_2}{s}, & a_2 &= \frac{\partial u_2}{\partial v_1} = \frac{h_2 - h_8 u_2}{s}, \\ a_3 &= \frac{\partial v_2}{\partial u_1} = \frac{h_4 - h_7 v_2}{s}, & a_4 &= \frac{\partial v_2}{\partial v_1} = \frac{h_5 - h_8 v_2}{s}, \end{aligned} \quad (1)$$

where  $u_i$  and  $v_i$  are the directions in the  $i$ th image ( $i \in \{1, 2\}$ ) and  $s = u_1 h_7 + v_1 h_8 + h_9$  is the projective depth.

### Fundamental Matrix

$$\mathbf{F} = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}$$

is a  $3 \times 3$  transformation matrix ensuring the so-called epipolar constraint  $\mathbf{p}_2^T \mathbf{F} \mathbf{p}_1 = 0$  for rigid scenes. Since its scale is arbitrary and  $\det(\mathbf{F}) = 0$ ,  $\mathbf{F}$  has seven degrees-of-freedom (DoF).

The relationship of affine correspondences and epipolar geometry is defined in (Barath et al., 2017) as follows:

$$\mathbf{A}^{-T} (\mathbf{F}^T \mathbf{p}_2)_{(1:2)} = (\mathbf{F} \mathbf{p}_1)_{(1:2)}, \quad (2)$$

where operator  $\mathbf{v}_{(i:j)}$  selects a sub-vector consisting of the elements of vector  $\mathbf{v}$  from the  $i$ th to the  $j$ th. Vector  $\mathbf{F}^T \mathbf{p}_2$  is basically the normal of the epipolar line on which point  $\mathbf{p}_1$  lies in the first image, and  $\mathbf{F} \mathbf{p}_1$  is the same for  $\mathbf{p}_2$ . Expanding this formula leads to a system of two linear, homogeneous equations as follows:

$$(u_2 + a_1 u_1) f_1 + a_1 v_1 f_2 + a_1 f_3 + (v_2 + a_3 u_2) f_4 + a_3 v_1 f_5 + a_3 f_6 + f_7 = 0, \quad (3)$$

$$a_2 u_1 f_1 + (u_2 + a_2 v_1) f_2 + a_2 f_3 + a_4 u_1 f_4 + (v_2 + a_4 v_1) f_5 + a_4 f_6 + f_8 = 0. \quad (4)$$

Thus each affine correspondence reduces the degrees-of-freedom by three. Having three of them are enough for the estimation. These properties will help us to recover an approximate fundamental matrix from six rotation invariant feature correspondences.

### 3 FUNDAMENTAL MATRIX ESTIMATION

In this section, we first propose two constraints reformulating the one given in (Barath et al., 2017). Comparing to the original one, the proposed constraints are separable into two, geometrically interpretable (rotation and scaling), transformations. We simplify the affine transformation model using the given rotations and this simplified model is then used to approximate the fundamental matrix.

#### 3.1 Geometric Constraints

As it was shown before, an affine correspondence yields three linear equations. However, in our case, not a full affine correspondence is given but a part of it: the point coordinates and a rotation. Therefore, assume point pair  $\mathbf{p}_1$ ,  $\mathbf{p}_2$  and the related angle,  $\alpha \in [0, 2\pi)$ , to be known in two images.

To exploit solely the rotation from the local affine transformation Eqs. 2 have to be reformulated. First replace  $\mathbf{F}^T \mathbf{p}_2$  and  $\mathbf{F} \mathbf{p}_1$  by the related normals as

$$\mathbf{A}^{-T} \mathbf{n}_1 = -\mathbf{n}_2.$$

It means that the coordinates of the line normal in the first image  $\mathbf{n}_1 = [n_{u,1} \ n_{v,1}]^T$  must be mapped to its coordinates  $\mathbf{n}_2 = [n_{u,2} \ n_{v,2}]^T$  in the second one by  $\mathbf{A}^{-T}$ . This relationship can be separated into orientation and scale components.

The **orientation constraint** states that  $\mathbf{A}^{-T}$  must rotate the direction of  $\mathbf{n}_1$  into that of  $\mathbf{n}_2$  as follows:

$$\mathbf{A}^{-T} \mathbf{n}_1 \parallel \mathbf{n}_2,$$

This can be written as

$$\frac{(\mathbf{A}^{-T} \mathbf{n}_1)^T \mathbf{n}_2}{|\mathbf{A} \mathbf{n}_1| |\mathbf{n}_2|} = \cos(0) = 1.$$

In order to eliminate the length calculation ( $|\mathbf{A}^{-T} \mathbf{n}_1|$  and  $|\mathbf{n}_2|$ ), it is beneficial to require perpendicularity instead of parallelity. Thus the original equations are formulated equivalently in the following way:

$$\frac{(\mathbf{A}^{-T} \mathbf{n}_1)^T \mathbf{R}^{\pi/2} \mathbf{n}_2}{|\mathbf{A} \mathbf{n}_1| |\mathbf{R}^{\pi/2} \mathbf{n}_2|} = \cos(\pi/2) = 0, \quad (5)$$

where  $\mathbf{R}^{\pi/2}$  is an orthonormal 2D rotation matrix rotating by  $\pi/2$  radians and  $\mathbf{R}^{\pi/2} \mathbf{n}_2 (= \mathbf{v}_2 = [v_{u,2} \ v_{v,2}]^T)$

is basically the tangent direction of the epipolar line in the second image.  $\mathbf{R}^{\pi/2}$  is as follows:

$$\mathbf{R}^{\pi/2} = \begin{bmatrix} \cos(\frac{\pi}{2}) & -\sin(\frac{\pi}{2}) \\ \sin(\frac{\pi}{2}) & \cos(\frac{\pi}{2}) \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Since for requiring Eq. 5 to be zero, the lengths do not have to be estimated. Thus Eq. 5 becomes

$$(\mathbf{A}^{-T} \mathbf{n}_1)^T \mathbf{R}^{\pi/2} \mathbf{n}_2 = 0.$$

To avoid inversion this can be written as follows:

$$(\mathbf{R}^{\pi/2} \mathbf{n}_1)^T (\mathbf{A}^T \mathbf{n}_2) = 0.$$

This formula leads to the following equation

$$(a_1 n_{u,2} + a_2 n_{v,2}) v_{u,1} + (a_3 n_{u,2} + a_4 n_{v,2}) v_{v,1} = 0.$$

Remember that  $\mathbf{n}_2$  and  $\mathbf{v}_1$  are computed from the fundamental matrix. They are as

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{R}^{\pi/2} \mathbf{n}_1 = \mathbf{R}^{\pi/2} (\mathbf{F}^T \mathbf{p}_2)_{(1:2)} = \\ \mathbf{R}^{\pi/2} \begin{bmatrix} f_1 u_2 + f_4 v_2 + f_7 \\ f_2 u_2 + f_5 v_2 + f_8 \end{bmatrix} &= \begin{bmatrix} f_2 u_2 + f_5 v_2 + f_8 \\ -f_1 u_2 - f_4 v_2 - f_7 \end{bmatrix}, \end{aligned}$$

and

$$\mathbf{n}_2 = (\mathbf{F} \mathbf{p}_1)_{(1:2)} = \begin{bmatrix} f_1 u_1 + f_2 v_1 + f_3 \\ f_4 u_1 + f_5 v_1 + f_6 \end{bmatrix}.$$

The final formula encapsulating the orientation constraint is the following:

$$\begin{aligned} (-u_2 f_2 - v_2 f_5 - f_8)(a_1(u_1 f_1 + v_1 f_2 + f_3) + \\ (u_2 f_1 + v_2 f_4 + f_7)(a_2(u_1 f_1 + v_1 f_2 + f_3) + \\ a_3(u_1 f_4 + v_1 f_5 + f_6)) + a_4(u_1 f_4 + v_1 f_5 + f_6)) = 0 \end{aligned} \quad (6)$$

The **scale constraint** states that the length of the transformed normal in the first image must be the inverse of that in the second one. Due to  $\mathbf{A}^{-T} \mathbf{n}_1 = -\mathbf{n}_2$ ,  $|\mathbf{n}_1|$  equals to  $|\mathbf{A}^T \mathbf{n}_2|$ . Therefore,

$$n_{u,1}^2 + n_{v,1}^2 = (a_1 n_{u,2} + a_2 n_{v,2})^2 + (a_3 n_{u,2} + a_4 n_{v,2})^2. \quad (7)$$

The final formula for the scale constraint (after rearranging Eq. 7 and substituting the elements of the fundamental matrix) is as follows:

$$\begin{aligned} (f_1 u_2 + f_4 v_2 + f_7)^2 + (f_2 u_2 + f_5 v_2 + f_8)^2 - \\ (a_1(f_1 u_1 + f_2 v_1 + f_3) + a_2(f_4 u_1 + f_5 v_1 + f_6))^2 - \\ (a_3(f_1 u_1 + f_2 v_1 + f_3) + a_4(f_4 u_1 + f_5 v_1 + f_6))^2 = 0 \end{aligned} \quad (8)$$

#### 3.2 Fundamental Matrix Estimation

Suppose that six point pairs  $\mathbf{p}_{1,i} = [u_{1,i} \ v_{1,i} \ 1]^T$ ,  $\mathbf{p}_{2,i} = [u_{2,i} \ v_{2,i} \ 1]^T$ , ( $i \in [1, 6]$ ) in their homogeneous form and the corresponding rotations  $\alpha_i$  are given. The task is to estimate fundamental matrix  $\mathbf{F}$  which is compatible with both the point coordinates and the given rotations. In order to reduce the

number of unknowns we first form a linear homogeneous equation system  $\mathbf{C}\mathbf{x} = 0$  from the point coordinates via the well-known formula,  $\mathbf{p}_{2,i}^T \mathbf{F} \mathbf{p}_{1,i} = 0$ . Vector  $\mathbf{x} = [f_1 \ f_2 \ f_3 \ f_4 \ f_5 \ f_6 \ f_7 \ f_8 \ f_9]^T$  consists of the unknown variables and coefficient matrix  $\mathbf{C} \in \mathbb{R}^{6 \times 9}$  is as follows:

$$\mathbf{C} = \begin{bmatrix} u_{1,1}u_{2,1} & v_{1,1}u_{2,1} & u_{2,1} & u_{1,1}v_{2,1} & v_{1,1}v_{2,1} & v_{2,1} & u_{1,1} & v_{1,1} & 1 \\ u_{1,6}u_{2,6} & v_{1,6}u_{2,6} & u_{2,6} & u_{1,6}v_{2,6} & v_{1,6}v_{2,6} & v_{2,6} & u_{1,6} & v_{1,6} & 1 \end{bmatrix}.$$

Since the null-space of  $\mathbf{C}$  is three-dimensional,  $\mathbf{F}$  can be calculated as their linear combination as

$$\mathbf{F} = \beta \mathbf{e} + \gamma \mathbf{g} + \delta \mathbf{h}, \quad (9)$$

where  $\beta$ ,  $\gamma$  and  $\delta$  are unknown scalars,  $\mathbf{e} = [e_1 \dots e_9]^T$ ,  $\mathbf{g} = [g_1 \dots g_9]^T$  and  $\mathbf{h} = [h_1 \dots h_9]^T$  are the null-vectors. Due to the scale ambiguity of  $\mathbf{F}$  one scale can be chosen to an arbitrary value, thus  $\delta = 1$  in our algorithm.

In order to exploit the given rotations we approximate the affine transformation model assuming that the orientation constraint (Eq. 8) can be satisfied by purely a rotation. Note that, without proof, this assumption holds for orthogonal projection. However, for perspective projection, the shear also affects Eq. 8. This approximation means that  $\mathbf{A}_i \approx \mathbf{R}^{\alpha_i}$ , where  $\mathbf{R}^{\alpha_i}$  is a 2D rotation matrix rotating by  $\alpha_i$  radians. We do not exploit the scale constraint, since no information is available about how  $\mathbf{A}$  scales the normal of the related epipolar line.

Replacing each  $f_i$  with  $\beta e_i + \gamma g_i + h_i$  in Eq. 8 leads to a multivariate polynomial equation (see Appendix 5 for details) with monomials  $[\beta^2 \ \gamma^2 \ \beta\gamma \ \beta \ \gamma \ 1]^T$ . Since we are given six rotations, it yields six polynomial equations. Considering each monomial as an independent variable the obtained inhomogeneous linear system  $\mathbf{D}\mathbf{h} = \mathbf{d}$  becomes solvable, where  $\mathbf{D}$  is the coefficient matrix,  $\mathbf{h} = [\beta^2 \ \gamma^2 \ \beta\gamma \ \beta \ \gamma]^T$  consists of the unknown variables and  $\mathbf{d}$  is the vector containing the inhomogeneous part of the equations. The final solution is obtained as  $\mathbf{h} = \mathbf{D}^\dagger \mathbf{d}$ , where  $\mathbf{D}^\dagger$  is the Moore-Penrose pseudo-inverse of matrix  $\mathbf{D}$ . The final fundamental matrix is calculated by substituting the obtained  $\beta$  and  $\gamma$  into Eq. 9. Note that it is beneficial to get the final scalars as  $\beta = \frac{\sqrt{h_1+h_4}}{2}$  and  $\gamma = \frac{\sqrt{h_2+h_5}}{2}$ . Due to the approximation we made,  $\mathbf{F}$  is not the exact fundamental matrix. However, it is good starting point for numerical optimization or, as we will demonstrate it later, for locally optimized robust estimators like LO-RANSAC (Chum et al., 2003).

**Robustness:** is achieved by normalizing the coefficients of each polynomial equation making their sum

one. It is also beneficial to apply the normalization technique of Hartley (Hartley, 1997) which is as follows: given a set  $\{(\mathbf{p}_{1,j}, \mathbf{p}_{2,j})\}_{j=1}^n$  of  $n$  point correspondences in their homogeneous forms. Normalizing transformation  $\mathbf{T}_i$  in the  $i$ th image ( $i \in \{1, 2\}$ ) is as follows:

$$\mathbf{T}_i = \begin{bmatrix} \sqrt{2}/d_i & 0 & 0 \\ 0 & \sqrt{2}/d_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\bar{u}_i \\ 0 & 1 & -\bar{v}_i \\ 0 & 0 & 1 \end{bmatrix}$$

where  $\bar{\mathbf{p}}_i = [\bar{u}_i, \bar{v}_i, 1]^T$  is the mean of the point set in the  $i$ th image and

$$d_i = \frac{1}{n} \sum_{j=1}^n \sqrt{(\mathbf{p}_i - \bar{\mathbf{p}})^T (\mathbf{p}_i - \bar{\mathbf{p}})} \quad (10)$$

is its average distance from the mean. Applying the normalizing transformations, the normalized correspondence set is as follows:  $\{(\hat{\mathbf{p}}_{1,j}, \hat{\mathbf{p}}_{2,j})\}_{j=1}^n = \{(\mathbf{T}_1 \mathbf{p}_{1,j}, \mathbf{T}_2 \mathbf{p}_{2,j})\}_{j=1}^n$ . After the estimation,  $\mathbf{F}$  is calculated from the normalized fundamental matrix  $\hat{\mathbf{F}}$  as follows:  $\mathbf{F} = \mathbf{T}_2^{-T} \hat{\mathbf{F}} \mathbf{T}_1^{-1}$ . Note that the normalizing transformations consist of a scale and a translation, therefore, the rotations of the features remain the same, they do not have to be modified.

### 3.3 Processing Time

The proposed method consists of two main steps: (i) the null-space computation of a matrix of size  $6 \times 9$ . (ii) Using the estimated null-vectors and the rotation components, a coefficient matrix of size  $6 \times 5$  is built and its null-space is computed. The average processing time of 100 runs of our C++ implementation using OpenCV was 0.03 milliseconds.

Combining *hypothesize-and-verify* robust estimators, like RANSAC (Fischler and Bolles, 1981), with the proposed method is beneficial since their processing time highly depends on the size of the minimal sample required for the estimation. Table 1 shows the theoretically needed iteration number of RANSAC combined with minimal methods (columns) on different outlier levels (rows). The confidence value was set to 95%. It can be seen that using the proposed 6-point algorithm leads to significant improvement in the processing time.

## 4 EXPERIMENTAL RESULTS

In this section, we will compare the proposed method with the eight- and seven-point algorithms on publicly available real datasets.

Table 1: Required theoretical iteration number of RANSAC (Fischler and Bolles, 1981) combined with minimal methods (columns) with confidence set to 95% on different outlier levels (rows).

Outl.	Confidence 95%		
	6	7	8
50%	<b>190</b>	382	765
80%	$\sim 10^4$	$\sim 10^5$	$\sim 10^6$
95%	$\sim 10^8$	$\sim 10^9$	$\sim 10^{10}$
99%	$\sim 10^{12}$	$\sim 10^{14}$	$\sim 10^{16}$

In order to overcome the approximative nature of the proposed *approximating six-point technique* we combined it with a recent variant of locally optimized RANSAC (Chum et al., 2003). We chose Graph-Cut RANSAC (Barath and Matas, 2017) (GC-RANSAC) as robust estimator since it can be considered as state-of-the-art and its source code is publicly available<sup>1</sup>. Briefly, it replaces the local optimization step of LO-RANSAC with graph-cut applied to the current best model. In the local optimization step, we used the normalized eight-point algorithm. Thus the approximated fundamental matrix is used only as an initial estimate to determine a set of inliers, then the obtained  $\mathbf{F}$  is refined iteratively exploiting a set of correspondences, i.e. the inliers. We used the same parameters as the authors proposed: the inlier-outlier threshold was set to 0.31, the iteration limit to 5000 and the weight of the spatial coherence term was 0.14.

We used the AdelaideRMF, Kusvod2, Multi-H, and Strecha datasets (see Fig. 2) to evaluate the proposed method on real world data. AdelaideRMF, Kusvod2 and Multi-H contains image pairs of size from  $455 \times 341$  to  $2592 \times 1944$  and manually annotated point correspondences (assigned to the outlier or inlier classes) for each pair. Since the reference points do not contain rotation components we detected and matched points applying ORB feature detector (Rublee et al., 2011). ORB features contain the orientation and the point coordinates.

Strecha dataset contains image sequences together with projection matrices. Each image is of resolution  $3072 \times 2048$ . The fundamental matrices are estimated for all possible image pairs in every sequence. Correspondences were obtained by ORB detector and the ground truth fundamental matrices were calculated from the given projection matrices (Hartley and Zisserman, 2003). All detected point pairs were considered as reference points for which the symmetric epipolar distance (Hartley and Zisserman, 2003) from the ground truth  $\mathbf{F}$  was smaller than 1.0 pixels. To discard not stable image pairs, the minimum reference point number was set to 10. Thus every image pair

for which less than 10 correspondences were closer to the ground truth  $\mathbf{F}$  than one pixels was not used in the evaluation.

We used the reference point sets to validate the estimated fundamental matrices. The reported geometric errors were computed as the mean symmetric epipolar distance as

$$\frac{1}{2} \sum_{(\mathbf{p}_1, \mathbf{p}_2) \in \mathcal{P}_R} \frac{\mathbf{F}\mathbf{p}_1}{\sqrt{(\mathbf{F}\mathbf{p}_1)_1^2 + (\mathbf{F}\mathbf{p}_1)_2^2}} + \frac{\mathbf{p}_2^T \mathbf{F}}{\sqrt{(\mathbf{p}_2^T \mathbf{F})_1^2 + (\mathbf{p}_2^T \mathbf{F})_2^2}}, \quad (11)$$

where  $\mathcal{P}_R$  is the set consisting of the reference point correspondences.

The competitor methods, i.e. the minimal solvers combined with GC-RANSAC, were the normalized eight- and seven-point algorithms<sup>2</sup>. In the least-squares model re-fitting step of GC-RANSAC, the normalized eight-point method was applied using the current inlier set.

Table 3 reports the mean result of 100 runs on every pair from the Strecha dataset. The first column denotes the name of the sequence, the second one is the number of the image pairs used – the ones for which more than 10 reference points were kept. The next two blocks, each consisting of three columns, shows the results of the methods if the confidence of GC-RANSAC is set to 99% (first block) and for a strict time limit (60 FPS; second block). The reported properties are the mean and median geometric errors of the estimated fundamental matrices (Eq. 11) w.r.t. the reference point sets, and the number of the samples, i.e. iterations, drawn by GC-RANSAC. It can be seen that for no time limit (first three columns), the seven-point algorithm obtains the most accurate results on average. This is not surprising since it is a consistent estimator (for no noise the error is zero) and “infinite” time is given to get the most accurate result. It can also be seen that if there is a strict time limit to achieve real time performance (last three columns), the proposed method yields the most accurate results.

Table 2 shows the mean results on AdelaideRMF, Kusvod2 and Multi-H datasets (first column) if the confidence is set to 99% (third – sixth column). The last three columns report the results if there is a strict 60 frames-per-second (FPS) time limit. The same property can be seen as for the Strecha dataset: (i) for no time limit, the best is the seven-point algorithm. (ii) For 1/60 FPS, the proposed approximating six-point method combined with GC-RANSAC leads to the most accurate fundamental matrix estimates.

<sup>2</sup>The implementation provided in OpenCV is used for the eight- and seven-point algorithms.

<sup>1</sup><https://github.com/danini/graph-cut-ransac>

Table 2: Fundamental matrix estimation on the (a) Kusvod2, (b) AdelaideRMF, and (c) Multi-H datasets applying GC-RANSAC (Barath and Matas, 2017) combined with minimal methods (second row). The number of the image pairs and the properties are written into the second and third columns. The results at 99% confidence are reported in the next three. The last three columns contain the results for a time limit set to 60 FPS, i.e. the run is interrupted after 1/60 secs. Values are computed as the means of 100 runs on each test pair. The mean and median errors (in pixels) of the estimated model w.r.t. the manually annotated inliers are written in each first and second rows; the required number of samples are reported in every third row.

			Confidence 99%			60 FPS		
Minimal methods			6	7	8	6	7	8
(a)	24	Avg Err (px)	<b>21.67</b>	23.78	47.67	<b>16.62</b>	20.18	45.38
		Med Err (px)	<b>22.45</b>	24.72	43.50	<b>13.88</b>	21.81	44.06
		Samples	<b>4992</b>	5000	5000	<b>74</b>	119	380
(b)	18	Avg Err (px)	4.63	<b>3.04</b>	6.72	4.00	<b>3.67</b>	7.63
		Med Err (px)	3.63	<b>2.33</b>	4.05	3.56	<b>2.82</b>	4.87
		Samples	<b>4982</b>	5000	5000	<b>74</b>	136	292
(c)	4	Avg Err (px)	<b>4.17</b>	4.45	11.26	<b>4.52</b>	4.59	12.18
		Med Err (px)	<b>4.46</b>	4.91	11.41	<b>4.15</b>	4.54	11.55
		Samples	<b>5000</b>	<b>5000</b>	<b>5000</b>	<b>51</b>	83	403
(all)		Avg Err (px)	<b>13.48</b>	13.98	28.48	<b>10.63</b>	12.36	27.72
		Med Err (px)	<b>10.18</b>	10.65	19.65	<b>7.20</b>	9.72	20.16
		Samples	<b>5000</b>	<b>5000</b>	<b>5000</b>	<b>72</b>	122	349

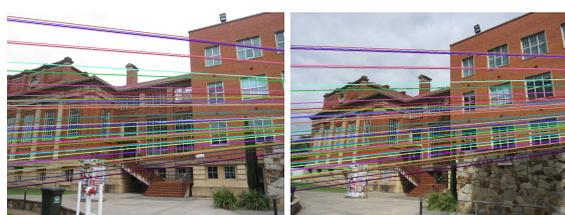
Table 3: Fundamental matrix estimation on the Strecha dataset applying GC-RANSAC (Barath and Matas, 2017) combined with minimal methods (second row). The first column contains the names of the sequences: (a) fountain-p11, (b) entry-p10, (c) castle-p19, (d) castle-p30, (e) herzjesus-p8, and (f) herzjesus-p25. The number of the image pairs and the properties are written into the second and third columns. The results at 99% confidence are reported in the next three. The last three columns contain the results for a time limit set to 60 FPS, i.e. the run is interrupted after 1/60 secs. Values are computed as the means of 100 runs on each test pair. The mean and median errors (in pixels) of the estimated model w.r.t. the manually annotated inliers are written in each first and second rows; the required number of samples are reported in every third row.

			Confidence 99%			60 FPS		
Minimal methods			6	7	8	6	7	8
(a)	19	Avg Err (px)	<b>1.93</b>	9.75	7.19	<b>1.84</b>	5.71	6.26
		Med Err (px)	<b>0.99</b>	3.42	2.78	<b>0.98</b>	1.64	3.12
		Samples	<b>4 394</b>	5 748	5 721	<b>279</b>	352	447
(b)	5	Avg Err (px)	30.15	<b>7.83</b>	13.67	<b>5.71</b>	7.04	41.90
		Med Err (px)	4.64	<b>1.74</b>	3.08	<b>1.56</b>	2.27	39.71
		Samples	<b>4 981</b>	5 121	5 004	<b>134</b>	183	284
(c)	21	Avg Err (px)	<b>2.43</b>	3.98	12.62	<b>5.42</b>	7.16	20.00
		Med Err (px)	<b>1.77</b>	2.90	4.62	<b>1.69</b>	5.46	9.87
		Samples	<b>4 575</b>	5 355	5 613	<b>210</b>	348	436
(d)	51	Avg Err (px)	9.24	<b>5.11</b>	10.73	<b>3.33</b>	6.29	9.82
		Med Err (px)	2.92	<b>1.89</b>	4.24	<b>1.32</b>	2.11	4.49
		Samples	<b>5 669</b>	6 306	6 339	<b>200</b>	311	392
(e)	16	Avg Err (px)	2.34	<b>1.77</b>	16.51	<b>2.85</b>	3.11	8.50
		Med Err (px)	<b>0.81</b>	1.50	4.75	2.05	<b>1.73</b>	2.65
		Samples	<b>6 492</b>	6 708	6 757	<b>199</b>	249	382
(f)	45	Avg Err (px)	5.47	<b>4.31</b>	10.50	<b>1.24</b>	2.27	6.46
		Med Err (px)	<b>1.71</b>	1.84	2.57	<b>1.24</b>	1.32	2.71
		Samples	<b>6 305</b>	6 708	6 758	<b>179</b>	240	361
(all)	157	Avg Err (px)	6.33	<b>5.04</b>	11.17	<b>3.62</b>	4.80	12.95
		Med Err (px)	1.93	<b>1.87</b>	3.49	<b>1.36</b>	2.11	3.96
		Samples	<b>5 613</b>	6 199	6 252	<b>200</b>	280	387

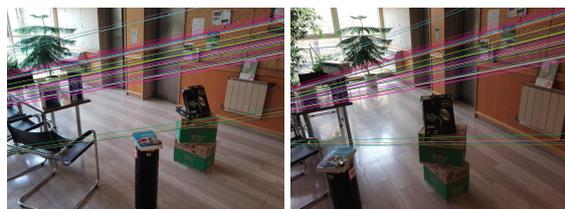
Fig. 2 shows example image pairs from each dataset with the epipolar lines of 50 random inliers. It can be seen, that the results seem good: the epipolar lines go through the same pixels in the first (left) and second (right) images.

## 5 CONCLUSION

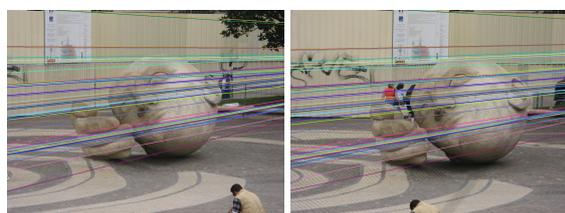
In this paper, we proposed a method for approximating the fundamental matrix between two non-calibrated views from six rotation invariant feature



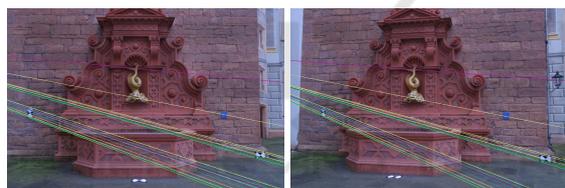
(a) AdelaideRMF



(b) Multi-H



(c) Kusvod2



(d) Strecha

Figure 2: The results of the proposed method combined with Graph-Cut RANSAC. An image pair from each dataset with the corresponding epipolar lines of 50 random inliers drawn by colors.

correspondences. The method is solved through the null-space computation of two matrices of sizes  $6 \times 9$  and  $6 \times 5$ , thus achieving fast calculation, i.e. around 0.03 milliseconds in C++. Due to the reduced number of required samples, the *approximating six-point algorithm* combined with locally optimized RANSAC obtains results superior to the state-of-the-art if a strict time limit is given.

## ACKNOWLEDGEMENTS

This research was supported by the Hungarian Scientific Research Fund (No. OTKA/NKFIH 120499), the Hungarian National Research, Development and Innovation Office under the grant VKSZ 14-1-2015-0072, and the European Union, co-financed by

the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001).

## REFERENCES

- Barath, D. (2017). P-HAF: Homography estimation using partial local affine frames. In *VISAPP*.
- Barath, D. and Matas, J. (2017). Graph-cut ransac. *ArXiv preprint arXiv:1706.00984*.
- Barath, D., Toth, T., and Hajder, L. (2017). A minimal solution for two-view focal-length estimation using two affine correspondences. In *Conference on Computer Vision and Pattern Recognition*.
- Batra, D., Nabbe, B., and Hebert, M. An alternative formulation for five point relative pose problem. In *Workshop on Motion and Video Computing*.
- Bentolila, J. and Francos, J. M. (2014). Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding*.
- Chum, O., Matas, J., and Kittler, J. (2003). Locally optimized ransac. In *Joint Pattern Recognition Symposium*.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*.
- Hartley, R. and Li, H. (2012). An efficient hidden variable approach to minimal-case camera motion estimation. *Pattern Analysis and Machine Intelligence*.
- Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.
- Hartley, R. I. (1997). In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence*.
- Kukelova, Z., Bujnak, M., and Pajdla, T. (2008). Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *British Machine Vision Conference*.
- Li, H. (2006). A simple solution to the six-point two-view focal-length problem. In *European Conference on Computer Vision*.
- Li, H. and Hartley, R. (2006). Five-point motion estimation made easy. In *International Conference on Pattern Recognition*.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer vision*.
- Matas, J., Chum, O., Urban, M., and Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*.
- Molnár, J. and Chetverikov, D. (2014). Quadratic transformation for planar mapping of implicit surfaces. *Journal of Mathematical Imaging and Vision*.
- Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence*.
- Perdoch, M., Matas, J., and Chum, O. (2006). Epipolar geometry from two correspondences. In *International Conference on Pattern Recognition*.
- Raposo, C. and Barreto, J. P. (2016). Theory and practice of structure-from-motion using affine correspondences. In *Computer Vision and Pattern Recognition*.

- Ruble, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE.
- Scaramuzza, D. (2011). 1-point-nistac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International Journal of Computer Vision*.
- Stewénius, H., Nistér, D., Kahl, F., and Schaffalitzky, F. (2008). A minimal solution for relative pose with unknown focal length. *Image Vision Computing*.
- Torii, A., Kukulova, Z., Bujnak, M., and Pajdla, T. (2010). The six point algorithm revisited. In *Asian Conference on Computer Vision*.

## APPENDIX

### Formulas for the Orientation Constraint

Replacing each  $f_i$  with  $\beta e_i + \gamma g_i + h_i$  in Eq. 8 leads to a multivariate polynomial equation with monomials  $[\beta^2 \ \gamma^2 \ \beta\gamma \ \beta \ \gamma \ 1]^T$ . The coefficients regarding to each monomial are shown in Figure 3.

$$\begin{aligned}
 \beta^2 & : -a_3b_5^2v_1v_2 + a_4b_4b_5v_1v_2 - a_1b_2b_5v_1v_2 + a_2b_2b_4v_1v_2 - a_3b_4b_5u_1v_2 - a_1b_1b_5u_1v_2 + a_4b_4^2u_1v_2 + a_2b_1b_4u_1v_2 - a_3b_5b_6v_2 + \\
 & a_4b_4b_6v_2 - a_1b_3b_5v_2 + a_2b_3b_4v_2 - a_3b_2b_5u_2v_1 + a_4b_1b_5u_2v_1 - a_1b_2^2u_2v_1 + a_2b_1b_2u_2v_1 - a_3b_5b_8v_1 - a_1b_2b_8v_1 + a_3b_2b_6u_2 + \\
 & a_4b_1b_6u_2 - a_1b_2b_3u_2 + a_4b_5b_7v_1 + a_2b_2b_7v_1 - a_3b_2b_4u_1u_2 + a_4b_1b_4u_1u_2 - a_1b_1b_2u_1u_2 + a_2b_1^2u_1u_2 - a_2b_1b_3u_2 - a_3b_4b_8u_1 - \\
 & a_1b_1b_8u_1 + a_4b_4b_7u_1 + a_2b_1b_7u_1 - a_3b_6b_8 - a_1b_3b_8 + a_4b_6b_7 + a_2b_3b_7 \\
 \gamma^2 & : -a_3c_5^2v_1v_2 + a_4c_4c_5v_1v_2 - a_1c_2c_5v_1v_2 + a_2c_2c_4v_1v_2 - a_3c_4c_5u_1v_2 - a_1c_1c_5u_1v_2 + a_4c_4^2u_1v_2 + a_2c_1c_4u_1v_2 - a_3c_5c_6v_2 + \\
 & a_4c_4c_6v_2 - a_1c_3c_5v_2 + a_2c_3c_4v_2 - a_3c_2c_5u_2v_1 + a_4c_1c_5u_2v_1 - a_1c_2^2u_2v_1 + a_2c_1c_2u_2v_1 - a_3c_5c_8v_1 - a_1c_2c_8v_1 + \\
 & a_4c_5c_7v_1 + a_2c_2c_7v_1 - a_3c_2c_4u_1u_2 + a_4c_1c_4u_1u_2 - a_1c_1c_2u_1u_2 + a_2c_1^2u_1u_2 + a_3c_2c_6u_2 + a_4c_1c_6u_2 - a_1c_2c_3u_2 + \\
 & a_2c_1c_3u_2 - a_3c_4c_8u_1 - a_1c_1c_8u_1 + a_4c_4c_7u_1 + a_2c_1c_7u_1 - a_3c_6c_8 - a_1c_3c_8 + a_4c_6c_7 + a_2c_3c_7 \\
 \beta\gamma & : -2a_3b_5c_5v_1v_2 + a_4b_4c_5v_1v_2 - a_1b_2c_5v_1v_2 + a_4b_5c_4v_1v_2 + a_2b_2c_4v_1v_2 - a_1b_5c_2v_1v_2 + a_2b_4c_2v_1v_2 - a_3b_4c_5u_1v_2 - a_1b_1c_5u_1v_2 - \\
 & a_3b_5c_4u_1v_2 + 2a_4b_4c_4u_1v_2 + a_2b_1c_4u_1v_2 - a_1b_5c_1u_1v_2 + a_2b_4c_1u_1v_2 - a_3b_5c_6v_2 + a_4b_4c_6v_2 - a_3b_6c_5v_2 - a_1b_3c_5v_2 + \\
 & a_4b_6c_4v_2 + a_2b_3c_4v_2 - a_1b_5c_3v_2 + a_2b_4c_3v_2 - a_3b_2c_5u_2v_1 + a_4b_1c_5u_2v_1 - a_3b_5c_2u_2v_1 - 2a_1b_2c_2u_2v_1 + \\
 & a_2b_1c_2u_2v_1 + a_4b_5c_1u_2v_1 + a_2b_2c_1u_2v_1 - a_3b_5c_8v_1 - a_1b_2c_8v_1 + a_4b_5c_7v_1 + a_2b_2c_7v_1 - a_3b_8c_5v_1 + a_4b_7c_5v_1 - a_1b_8c_2v_1 + \\
 & a_2b_7c_2v_1 - a_3b_2c_4u_1u_2 + a_4b_1c_4u_1u_2 - a_3b_4c_2u_1u_2 - a_1b_1c_2u_1u_2 + a_4b_4c_1u_1u_2 - a_1b_2c_1u_1u_2 + 2a_2b_1c_1u_1u_2 - a_3b_2c_6u_2 + \\
 & a_4b_1c_6u_2 - a_1b_2c_3u_2 + a_2b_1c_3u_2 - a_3b_6c_2u_2 - a_1b_3c_2u_2 + a_4b_6c_1u_2 + a_2b_3c_1u_2 - a_3b_4c_8u_1 - a_1b_1c_8u_1 + a_4b_4c_7u_1 + a_2b_1c_7u_1 - \\
 & a_3b_8c_4u_1 + a_4b_7c_4u_1 - a_1b_8c_1u_1 + a_2b_7c_1u_1 - a_3b_6c_8 - a_1b_3c_8 + a_4b_6c_7 + a_2b_3c_7 - a_3b_8c_6 + a_4b_7c_6 - a_1b_8c_3 + a_2b_7c_3 \\
 \beta & : -2a_3b_5d_5v_1v_2 + a_4b_4d_5v_1v_2 - a_1b_2d_5v_1v_2 + a_4b_5d_4v_1v_2 + a_2b_2d_4v_1v_2 - a_1b_5d_2v_1v_2 + a_2b_4d_2v_1v_2 - a_3b_4d_5u_1v_2 - a_1b_1d_5u_1v_2 - \\
 & a_3b_5d_4u_1v_2 + 2a_4b_4d_4u_1v_2 + a_2b_1d_4u_1v_2 - a_1b_5d_1u_1v_2 + a_2b_4d_1u_1v_2 - a_3b_5d_6v_2 + a_4b_4d_6v_2 - a_3b_6d_5v_2 - a_1b_3d_5v_2 + a_4b_6d_4v_2 + \\
 & a_2b_1d_2u_2v_1 + a_4b_5d_1u_2v_1 + a_2b_3d_4v_2 - a_1b_5d_3v_2 + a_2b_4d_3v_2 - a_3b_2d_5u_2v_1 + a_4b_1d_5u_2v_1 - a_3b_5d_2u_2v_1 - 2a_1b_2d_2u_2v_1 + \\
 & a_2b_2d_1u_2v_1 - a_3b_5d_8v_1 - a_1b_2d_8v_1 + a_4b_5d_7v_1 + a_2b_2d_7v_1 - a_3b_8d_5v_1 + a_4b_7d_5v_1 - a_1b_8d_2v_1 + a_2b_7d_2v_1 - a_3b_2d_4u_1u_2 + \\
 & a_4b_1d_4u_1u_2 - a_3b_4d_2u_1u_2 - a_1b_1d_2u_1u_2 + a_4b_4d_1u_1u_2 - a_1b_2d_1u_1u_2 + 2a_2b_1d_1u_1u_2 - a_3b_2d_6u_2 + a_4b_1d_6u_2 - a_1b_2d_3u_2 + \\
 & a_2b_1d_3u_2 - a_3b_6d_2u_2 - a_1b_3d_2u_2 + a_4b_6d_1u_2 + a_2b_3d_1u_2 - a_3b_4d_8u_1 - a_1b_1d_8u_1 + a_4b_4d_7u_1 + a_2b_1d_7u_1 - a_3b_8d_4u_1 + a_4b_7d_4u_1 - \\
 & a_1b_8d_1u_1 + a_2b_7d_1u_1 - a_3b_6d_8 - a_1b_3d_8 + a_4b_6d_7 + a_2b_3d_7 - a_3b_8d_6 + a_4b_7d_6 - a_1b_8d_3 + a_2b_7d_3 \\
 \gamma & : -2a_3c_5d_5v_1v_2 + a_4c_4d_5v_1v_2 - a_1c_2d_5v_1v_2 + a_4c_5d_4v_1v_2 + a_2c_2d_4v_1v_2 - a_1c_5d_2v_1v_2 + a_2c_4d_2v_1v_2 - a_3c_4d_5u_1v_2 - a_1c_1d_5u_1v_2 - \\
 & a_3c_5d_4u_1v_2 + 2a_4c_4d_4u_1v_2 + a_2c_1d_4u_1v_2 - a_1c_5d_1u_1v_2 + a_2c_4d_1u_1v_2 - a_3c_5d_6v_2 + a_4c_4d_6v_2 - a_3c_6d_5v_2 - a_1c_3d_5v_2 + a_4c_6d_4v_2 + \\
 & a_2c_3d_4v_2 - a_1c_5d_3v_2 + a_2c_4d_3v_2 - a_3c_2d_5u_2v_1 + a_4c_1d_5u_2v_1 - a_3c_5d_2u_2v_1 - 2a_1c_2d_2u_2v_1 + a_2c_1d_2u_2v_1 + a_4c_5d_1u_2v_1 + \\
 & a_2c_2d_1u_2v_1 - a_3c_5d_8v_1 - a_1c_2d_8v_1 + a_4c_5d_7v_1 + a_2c_2d_7v_1 - a_3c_8d_5v_1 + a_4c_7d_5v_1 - a_1c_8d_2v_1 + a_2c_7d_2v_1 - a_3c_2d_4u_1u_2 + \\
 & a_4c_1d_4u_1u_2 - a_3c_4d_2u_1u_2 - a_1c_1d_2u_1u_2 + a_4c_4d_1u_1u_2 - a_1c_2d_1u_1u_2 + 2a_2c_1d_1u_1u_2 - a_3c_2d_6u_2 + a_4c_1d_6u_2 - a_1c_2d_3u_2 + \\
 & a_2c_1d_3u_2 - a_3c_6d_2u_2 - a_1c_3d_2u_2 + a_4c_6d_1u_2 + a_2c_3d_1u_2 - a_3c_4d_8u_1 - a_1c_1d_8u_1 + a_4c_4d_7u_1 + a_2c_1d_7u_1 - a_3c_8d_4u_1 + \\
 & a_4c_7d_4u_1 - a_1c_8d_1u_1 + a_2c_7d_1u_1 - a_3c_6d_8 - a_1c_3d_8 + a_4c_6d_7 + a_2c_3d_7 - a_3c_8d_6 + a_4c_7d_6 - a_1c_8d_3 + a_2c_7d_3 \\
 1 & : -a_3d_5^2v_1v_2 + a_4d_4d_5v_1v_2 - a_1d_2d_5v_1v_2 + a_2d_2d_4v_1v_2 - a_3d_4d_5u_1v_2 - a_1d_1d_5u_1v_2 + a_4d_4^2u_1v_2 + a_2d_1d_4u_1v_2 - a_3d_5d_6v_2 + \\
 & a_4d_4d_6v_2 - a_1d_3d_5v_2 + a_2d_3d_4v_2 - a_3d_2d_5u_2v_1 + a_4d_1d_5u_2v_1 - a_1d_2^2u_2v_1 + a_2d_1d_2u_2v_1 - a_3d_5d_8v_1 - a_1d_2d_8v_1 + a_4d_5d_7v_1 + \\
 & a_2d_2d_7v_1 - a_3d_2d_4u_1u_2 + a_4d_1d_4u_1u_2 - a_1d_1d_2u_1u_2 + a_2d_1^2u_1u_2 - a_3d_2d_6u_2 + a_4d_1d_6u_2 - a_1d_2d_3u_2 + a_2d_1d_3u_2 - a_3d_4d_8u_1 - \\
 & a_1d_1d_8u_1 + a_4d_4d_7u_1 + a_2d_1d_7u_1 - a_3d_6d_8 - a_1d_3d_8 + a_4d_6d_7 + a_2d_3d_7
 \end{aligned}$$

Figure 3: The coefficient for each monomial in the polynomial equation to which the orientation constraint lead (Eq. 8).