Homography from two orientation- and scale-covariant features

Daniel Barath[1,2] Zuzana Kukelova[1]
[1] VRG, Department of Cybernetics, Czech Technical University in Prague, Czech Republic
[2] Machine Perception Research Laboratory, MTA SZTAKI, Budapest, Hungary
barath.daniel@sztaki.mta.hu

Abstract

This paper proposes a geometric interpretation of the angles and scales which the orientation- and scale-covariant feature detectors, e.g. SIFT, provide. Two new general constraints are derived on the scales and rotations which can be used in any geometric model estimation tasks. Using these formulas, two new constraints on homography estimation are introduced. Exploiting the derived equations, a solver for estimating the homography from the minimal number of two correspondences is proposed. Also, it is shown how the normalization of the point correspondences affects the rotation and scale parameters, thus achieving numerically stable results. Due to requiring merely two feature pairs, robust estimators, e.g. RANSAC, do significantly fewer iterations than by using the four-point algorithm. When using covariant features, e.g. SIFT, the information about the scale and orientation is given at no cost. The proposed homography estimation method is tested in a synthetic environment and on publicly available real-world datasets.

1. Introduction

This paper addresses the problem of interpreting, in a geometrically justifiable manner, the rotation and scale which the orientation- and scale-covariant feature detectors, e.g. SIFT [22] or SURF [10], provide. Then, by exploiting these new constraints, we involve all the obtained parameters of the SIFT features (i.e. the point coordinates, angle, and scale) into the homography estimation procedure. In particular, we are interested in the minimal case, to estimate a homography from solely two correspondences.

Nowadays, a number of algorithms exist for estimating or approximating geometric models, e.g. homographies, using affine-covariant features. A technique, proposed by Perdoch et al. [29], approximates the epipolar geometry from one or two affine correspondences by converting them to point pairs. Bentolila and Francos [11] proposed a solution for estimating the fundamental matrix using three affine features. Raposo et al. [32, 31] and Barath et al. [6] showed that two correspondences are enough for estimating the relative camera motion. Moreover, two feature pairs are enough for solving the semi-calibrated case, i.e. when the objective is to find the essential matrix and a common unknown focal length [9]. Also, homographies can be estimated from two affine correspondences [17], and, in case of known epipolar geometry, from a single correspondence [5]. There is a one-to-one relationship between local affine transformations and surface normals [17, 8]. Pritts et al. [30] showed that the lens distortion parameters can be retrieved using affine features. Affine correspondences encode higher-order information about the scene geometry. This is the reason why the previously mentioned algorithms solve geometric estimation problems exploiting fewer features than point correspondence-based methods. This implies nevertheless their major drawback: obtaining affine features accurately (e.g. by Affine SIFT [28], MODS [26], Hessian-Affine, or Harris-Affine [24] detectors) is time-consuming and, thus, is barely doable in time-sensitive applications.

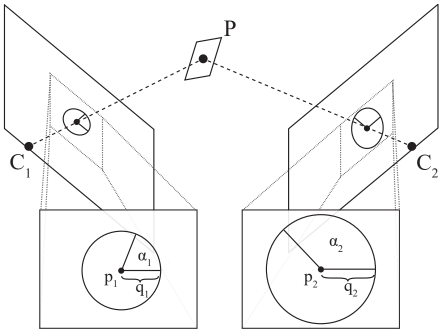


Figure 1: Visualization of the orientation- and scale-covariant features. Point $\mathbf{P}$ and the surrounding patch projected into cameras $\mathbf{C}_1$ and $\mathbf{C}_2$. A window showing the projected points $\mathbf{p}_1 = [u_1 \; v_1 \; 1]^\mathrm{T}$ and $\mathbf{p}_2 = [u_2 \; v_2 \; 1]^\mathrm{T}$ are cut out and enlarged. The rotation of the feature in the ith image is α_i and the size is q_i ($i \in \{1, 2\}$). The scaling from the 1st to the 2nd image is calculated as $q = q_2/q_1$.

Most of the widely-used feature detectors provide parts of the affine feature. For instance, there are detectors obtaining oriented features, e.g. ORB [33], or there are ones providing also the scales, e.g. SIFT [22] or SURF [10]. Exploiting this additional information is a well-known approach in, for example, wide-baseline matching [23, 26]. Yet, the first papers [1, 2, 3, 25, 4] involving them into geometric model estimation were published just in the last few years. In [25], the feature orientations are involved directly in the essential matrix estimation. In [1], the fundamental matrix is assumed to be a priori known and an algorithm is proposed for approximating a homography exploiting the rotations and scales of two SIFT correspondences. The approximative nature comes from the assumption that the scales along the axes are equal to the SIFT scale and the shear is zero. In general, these assumptions do not hold. The method of [2] approximates the fundamental matrix by enforcing the geometric constraints of affine correspondences on the epipolar lines. Nevertheless, due to using the same affine model as in [1], the estimated epipolar geometry is solely an approximation. In [3], a two-step procedure is proposed for estimating the epipolar geometry. First, a homography is obtained from three oriented features. Finally, the fundamental matrix is retrieved from the homography and two additional correspondences. Even though this technique considers the scales and shear as unknowns, thus estimating the epipolar geometry instead of approximating it, the proposed decomposition of the affine matrix is not justified theoretically. Therefore, the geometric interpretation of the feature rotations is not provably valid. A recently published paper [4] proposes a way of recovering full affine correspondences from the feature rotation, scale, and the fundamental matrix. Applying this method, a homography is estimated from a single correspondence in case of known epipolar geometry. Still, the decomposition of the affine matrix is ad hoc, and is, therefore, not a provably valid interpretation of the SIFT rotations and scales. Moreover, in practice, the assumption of the known epipolar geometry restricts the applicability of the method.

The contributions of this paper are: (i) we provide a geometrically valid way of interpreting orientation- and scale-covariant features approaching the problem by differential geometry. (ii) Building on the derived formulas, we propose two general constraints which hold for covariant features. (iii) These constraints are then used to derive two new formulas for homography estimation and (iv), based on these equations, a solver is proposed for estimating a homography matrix from two orientation- and scale-covariant feature correspondences. This additional information, i.e. the scale and rotation, is given at no cost when using most of the widely-used feature detectors, e.g. SIFT or SURF. It is validated both in a synthetic environment and on more than 10 000 publicly available real image pairs that the solver accurately recovers the homography matrix. Benefiting from the number of correspondences required, robust estimation, e.g. by GC-RANSAC [7], is two orders of magnitude faster than by combining it with the standard techniques, e.g. four-point algorithm [16].

2. Theoretical background

Affine correspondence $(\mathbf{p}_1, \mathbf{p}_2, \mathbf{A})$ is a triplet, where $\mathbf{p}_1 = [u_1 \; v_1 \; 1]^\mathrm{T}$ and $\mathbf{p}_2 = [u_2 \; v_2 \; 1]^\mathrm{T}$ are a corresponding homogeneous point pair in two images and $\mathbf{A}$ is a 2×2 linear transformation which is called *local affine transformation*. Its elements in a row-major order are: a_1, a_2, a_3, and a_4. To define $\mathbf{A}$, we use the definition provided in [27] as it is given as the first-order Taylor-approximation of the 3D → 2D projection functions. For perspective cameras, the formula for $\mathbf{A}$ is the first-order approximation of the related *homography* matrix as follows:

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

where u_i and v_i are the directions in the ith image ($i \in \{1, 2\}$) and $s = u_1 h_7 + v_1 h_8 + h_9$ is the projective depth. The elements of $\mathbf{H}$ in a row-major order are: h_1, h_2, ..., h_9.

The relationship of an affine correspondence and a homography is described by six linear equations. Since an affine correspondence involves a point pair, the well-known equations (from $\mathbf{H}\mathbf{p}_1 \sim \mathbf{p}_2$) hold [16]. They are as follows:

$$\begin{array}{c} u_1 h_1 + v_1 h_2 + h_3 - u_1 u_2 h_7 - v_1 u_2 h_8 - u_2 h_9 = 0, \\ u_1 h_4 + v_1 h_5 + h_6 - u_1 v_2 h_7 - v_1 v_2 h_8 - v_2 h_9 = 0. \end{array} \quad (2)$$

After re-arranging (1), four additional linear constraints are obtained from $\mathbf{A}$ which are the following.

$$\begin{array}{c} h_1 - (u_2 + a_1 u_1) h_7 - a_1 v_1 h_8 - a_1 h_9 = 0, \\ h_2 - (u_2 + a_2 v_1) h_8 - a_2 u_1 h_7 - a_2 h_9 = 0, \\ h_4 - (v_2 + a_3 u_1) h_7 - a_3 v_1 h_8 - a_3 h_9 = 0, \\ h_5 - (v_2 + a_4 v_1) h_8 - a_4 u_1 h_7 - a_4 h_9 = 0. \end{array} \quad (3)$$

Consequently, an affine correspondence provides six linear equations for the elements of the related homography.

3. Affine transformation model

In this section, the interpretation of the feature scales and rotations are discussed. Two new constraints that relate the elements of the affine transformation to the feature scale and rotation are derived. These constraints are general, and they can be used for estimating different geometric models, e.g. homographies or fundamental matrices, using orientation- and scale-covariant features. In this paper, the two constraints are used to derive a solver for homography estimation from two correspondences. For the sake of simplicity, we use SIFT as an alias for all the orientation- and scale-covariant detectors. The formulas hold for all of them.

3.1. Interpretation of the SIFT output

Reflecting the fact that we are given a scale $q_i \in \mathbb{R}$ and rotation $\alpha_i \in [0, 2\pi)$ independently in each image ($i \in \{1, 2\}$; see Fig. 1), the objective is to define affine correspondence $\mathbf{A}$ as a function of them. For this problem, approaches were proposed in the recent past [3, 4]. None of them were nevertheless proven to be a valid interpretation.

To understand the SIFT output, we exploit the definition of affine correspondences proposed in [8]. In [8], $\mathbf{A}$ is defined as the multiplication of the Jacobians of the projection functions in the two images as follows:

$$\mathbf{A} = \mathbf{J}_2 \mathbf{J}_1^{-1}, \tag{4}$$

where $\mathbf{J}_1$ and $\mathbf{J}_2$ are the Jacobians of the 3D → 2D projection functions. Proof is in Appendix A. For the ith Jacobian, the following is a possible decomposition:

$$\mathbf{J}_i = \mathbf{R}_i \mathbf{U}_i = \begin{bmatrix} \cos(\alpha_i) & -\sin(\alpha_i) \\ \sin(\alpha_i) & \cos(\alpha_i) \end{bmatrix} \begin{bmatrix} q_{u,i} & w_i \\ 0 & q_{v,i} \end{bmatrix}, \tag{5}$$

where angle α_i is the rotation in the ith image, $q_{u,i}$ and $q_{v,i}$ are the scales along axes u and v, and w_i is the shear ($i \in \{1, 2\}$). Let us use the following notation: $c_i = \cos(\alpha_i)$ and $s_i = \sin(\alpha_i)$. The equation for the inverse matrix becomes

$$\mathbf{J}_i^{-1} = \frac{1}{c_i^2 q_{u,i} q_{v,i} + s_i^2 q_{u,i} q_{v,i}} \begin{bmatrix} s_i w_i + c_i q_{v,i} & s_i q_{v,i} - c_i w_i \\ -s_i q_{u,i} & c_i q_{u,i} \end{bmatrix}.$$

The denominator can be formulated as follows: $(c_i^2 + s_i^2) q_{u,i} q_{v,i}$, where $c_i^2 + s_i^2$ is a trigonometric identity and equals to one. After multiplying the matrices in (4), the following equations are given for the affine elements:

$$a_1 = \frac{c_2 q_{u,2}(s_1 w_1 + c_1 q_{v,1}) - s_1 q_{u,1}(c_2 w_2 - s_2 q_{v,2})}{q_{u,1} q_{v,1}} \tag{6}$$

$$a_2 = \frac{c_2 q_{u,2}(s_1 q_{v,1} - c_1 w_1) + c_1 q_{u,1}(c_2 w_2 - s_2 q_{v,2})}{q_{u,1} q_{v,1}} \tag{7}$$

$$a_3 = \frac{s_2 q_{u,2}(s_1 w_1 + c_1 q_{v,1}) - s_1 q_{u,1}(s_2 w_2 + c_2 q_{v,2})}{q_{u,1} q_{v,1}} \tag{8}$$

$$a_4 = \frac{s_2 q_{u,2}(s_1 q_{v,1} - c_1 w_1) + c_1 q_{u,1}(s_2 w_2 + c_2 q_{v,2})}{q_{u,1} q_{v,1}} \tag{9}$$

These formulas show how the affine elements relate to α_i, the scales along axes u and v and shears w_i.

In case of having orientation- and scale-covariant features, e.g. SIFT, the known parameters are the rotation α_i of the feature in the ith image and a uniform scale q_i. It can be easily seen that the scale q_i is interpreted as follows:

$$q_i = \det \mathbf{J}_i = q_{u,i} q_{v,i}. \tag{10}$$

Therefore, our goal is to derive constraints that relate affine elements of $\mathbf{A}$ to the orientations α_i and scales q_i of the features in the first and second images. We will derive such constraints by eliminating the scales along axes $q_{u,i}$ and $q_{v,i}$ and the shears w_i from equations (6)-(9). To do this, we use an approach based on the elimination ideal theory [13]. Elimination ideal theory is a classical algebraic method for eliminating variables from polynomials of several variables. This method was recently used in [21] for eliminating unknowns from equations that are not dependent on input measurements. Here, we use the method in a slightly different way. We first create the ideal I [13] generated by polynomials (6)-(9), polynomial (10) and trigonometric identities $c_i^2 + s_i^2 = 1$ for $i \in \{1, 2\}$. Note that here we consider all elements of these polynomials, including c_i and s_i, as unknowns. Then we compute generators of the elimination ideal $I_1 = I \cap \mathbb{C}[a_1, a_2, a_3, a_4, q_1, q_2, s_1, c_1, s_2, c_2]$ [13]. The generators of I_1 do not contain $q_{u,i}$, $q_{v,i}$ and w_i. The elimination ideal I_1 is generated by two polynomials:

$$q_1^2 a_2 a_3 - q_1^2 a_1 a_4 + q_1 q_2 = 0, \tag{11}$$

$$c_1 s_2 q_1 a_1 + s_1 s_2 q_1 a_2 - c_1 c_2 q_1 a_3 - c_2 s_1 q_1 a_4 = 0. \tag{12}$$

Generators (11)-(12) can be computed using a computer algebra system, e.g. `Macaulay2` [14]. The new constraints relate the elements of $\mathbf{A}$ to the scales and rotations of the features in both images. Note that both these equations can be divided by $q_1 \neq 0$. After this simplification, (11) corresponds to $\det \mathbf{A} = q_2/q_1 = q$ and equation (12) relates the rotations of the features to the elements of $\mathbf{A}$. The two new constraints are general, and they can be used for estimating different geometric models, e.g. homographies or fundamental matrices, using orientation- and scale-covariant detectors. Next, we use (11)-(12) to derive new constraints on a homography.

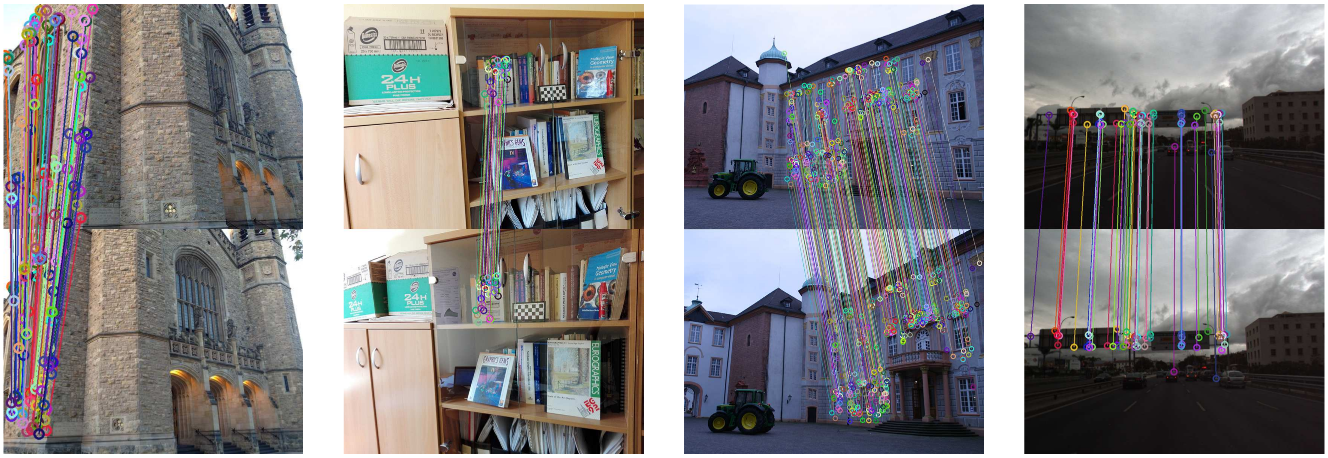
4. Homography from two correspondences

In this section, we derive new constraints that relate $\mathbf{H}$ to the feature scales and rotations in the two images. Then a solver is proposed to estimate $\mathbf{H}$ from two SIFT correspondences based on these new constraints. Finally, we discuss how the widely-used normalization of the point correspondences [15] affects the output of orientation- and scale-covariant detectors and subsequently the new constraints.

4.1. Homography and covariant features

First, we derive constraints that relate the homography $\mathbf{H}$ to the scales and rotations of the features in the first and second images. To do this, we combine constraints (11) and (12) derived in previous section with the constraints on the homography matrix (3).

Constraints (11) and (12) cannot be directly substituted into (3). However, we can use a similar approach as in the previous section for deriving (11) and (12). First, ideal J generated by six polynomials (3), (11) and (12) is constructed. Then the unknown elements of the affine transformation $\mathbf{A}$ are eliminated from the generators of J. We do this by computing the generators of $J_1 = J \cap$



(a) 553 iterations by 2SIFT and 8 615 by 4PT. Inlier ratio 0.38.

(b) 720 iterations by 2SIFT and 78 450 by 4PT. Inlier ratio 0.06.

(c) 169 iterations by 2SIFT and 573 by 4PT. Inlier ratio 0.22.

(d) 65 iterations by 2SIFT and 14 139 by 4PT. Inlier ratio 0.23.

Figure 2: Inliers of the estimated homographies (by 2SIFT) drawn to example image pairs. The numbers of iterations of GC-RANSAC [7] using the 4PT and proposed 2SIFT solvers; and the ground truth inlier ratios are reported in the captions.

$\mathbb{C}[h_1, \dots, h_9, u_1, v_1, u_2, v_2, q_1, q_2, s_1, c_1, s_2, c_2]$. The elimination ideal J_1 is generated by two polynomials:

$$h_8 u_2 s_1 s_2 + h_7 u_2 s_2 c_1 - h_8 v_2 s_1 c_2 - h_7 v_2 c_1 c_2 + \tag{13}$$
$$-h_2 s_1 s_2 - h_1 s_2 c_1 + h_5 s_1 c_2 + h_4 c_1 c_2 = 0,$$
$$h_7^2 u_1^2 q_2 + 2 h_7 h_8 u_1 v_1 q_2 + h_8^2 v_1^2 q_2 + h_5 h_7 u_2 q_1 + \tag{14}$$
$$-h_4 h_8 u_2 q_1 - h_2 h_7 v_2 q_1 + h_1 h_8 v_2 q_1 + 2 h_7 h_9 u_1 q_2 +$$
$$2 h_8 h_9 v_1 q_2 + h_2 h_4 q_1 - h_1 h_5 q_1 + h_9^2 q_2 = 0.$$

Polynomials (13) and (14) are new constraints that relate the homography matrix to the scales and rotations of the features in the first and second images. These constraints will help us for recovering $\mathbf{H}$ from two orientation- and scale-covariant feature correspondences.

4.2. 2-SIFT solver

Constraint (13) is linear in the elements of $\mathbf{H}$. For two SIFT correspondences, two such equations are given, which, together with the four equations for point correspondences (2), result in six homogeneous linear equations in the nine elements of $\mathbf{H}$. In matrix form, these equations are: $\mathbf{M}\,\mathbf{h} = \mathbf{0}$, where $\mathbf{M}$ is a 6×9 coefficient matrix and $\mathbf{h}$ contains the unknown homography elements. For two SIFT correspondences in two views, coefficient matrix $\mathbf{M}$ has a three-dimensional null space. Therefore, the homography matrix can be parameterized by two unknowns as

$$\mathbf{H} = x\,\mathbf{H}_1 + y\,\mathbf{H}_2 + \mathbf{H}_3, \tag{15}$$

where $\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3$ are created from the 3D null space of $\mathbf{M}$ and x and y are new unknowns. Now we can plug the parameterization (15) into constraint (14). For two SIFT correspondences, this results in two quadratic equations in two unknowns. Such equations have four solutions and they can be easily solved using e.g. the Gröbner basis or the resultant based method [13]. Here, we use the solver based on Gröbner basis method that can be created using the automatic generator [19]. This solver performs Gauss-Jordan elimination of a 6×10 template matrix which contains just monomial multiples of the two input equations. Then the solver extracts solutions to x and y from the eigenvectors of a 4×4 multiplication matrix that is extracted from the template matrix. Finally, up to four real solutions to $\mathbf{H}$ are computed by substituting solutions for x and y to (15).

Note that we do not know any degeneracies of the proposed solver which can occur in real life. For instance, the degeneracy of the four-point algorithm, i.e. the points are co-linear, is not a degenerate case for the 2SIFT solver.

4.3. Normalization of the affine parameters

The normalization of the point coordinates is a crucial step to increase the numerical stability of $\mathbf{H}$ estimation [15]. Suppose that we are given a 3×3 normalizing transformation $\mathbf{T}_i$ transforming the center of gravity of the point cloud in the ith image to the origin and its average distance from it to $\sqrt{2}$. The formula for normalizing $\mathbf{A}$ is as follows [6]:

$$\widehat{\mathbf{A}} = \mathbf{T}_2 \begin{bmatrix} \mathbf{A} & 0 \\ 0 & 1 \end{bmatrix} \mathbf{T}_1^{-1}, \tag{16}$$

where $\widehat{\mathbf{A}}$ is the normalized affinity. Matrix $\mathbf{T}_i$ transforms the points by translating them (last column) and applying a uniform scaling (diagonal). Due to the fact that the last column of $\mathbf{T}_i$ has no effect on the top-left 2×2 sub-matrix of the normalized affinity, the equation can be rewritten as follows: $\widehat{\mathbf{A}} = \mathrm{diag}(t_2, t_2)\ \mathbf{A}\ \mathrm{diag}(1/t_1, 1/t_1) = t_2/t_1 \mathbf{A}$,

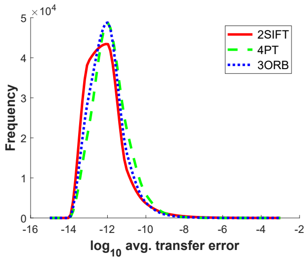


Figure 3: *Stability study.* The frequencies (100 000 runs; vertical axis) of $\log_{10}$ errors (horizontal) in the estimated homographies by the proposed (red), 4PT (green) and 3ORI (blue) methods.

where t_1 and t_2 are the scales of the normalizing transformations in the two images. Thus, for normalizing the affine transformation, it has to be multiplied by t_2/t_1.

The scaling factor affects constraint (11) which, for $\widehat{\mathbf{A}}$, has the form

$$t^2 q_1^2 \widehat{a}_2 \widehat{a}_3 - t^2 q_1^2 \widehat{a}_1 \widehat{a}_4 + q_1 q_2 = 0, \tag{17}$$

where $t = t_1/t_2$ and $\widehat{a}_i$ are elements of $\widehat{\mathbf{A}}$. Consequently constraint (14) for the normalized coordinates has the form

$$\begin{gathered} h_7^2 u_1^2 q_2 t^2 + 2 h_7 h_8 u_1 v_1 q_2 t^2 + h_8^2 v_1^2 q_2 t^2 + h_5 h_7 u_2 q_1 + \\ -h_4 h_8 u_2 q_1 - h_2 h_7 v_2 q_1 + h_1 h_8 v_2 q_1 + 2 h_7 h_9 u_1 q_2 t^2 + \\ 2 h_8 h_9 v_1 q_2 t^2 + h_2 h_4 q_1 - h_1 h_5 q_1 + h_9^2 q_2 t^2 = 0. \end{gathered} \tag{18}$$

Note that this normalization does not affect the structure of the derived 2SIFT solver. The only difference is that, for the normalized coordinates, the coefficients in the template matrix are multiplied by scale factor t as in (18).

5. Experimental results

In this section, we compare the proposed solver (2SIFT) with the widely-used normalized four-point (4PT) algorithm [16] and a method using three oriented features [3] (3ORI) for estimating the homography.

5.1. Computational complexity

First, we compare the computational complexity of the competitor algorithms, see Table 1. The first row consists of the major steps of each solver. For instance, 6 × 9 SVD + 6 × 6 QR + 4 × 4 EIG means, that the major steps are: the SVD decomposition of a 6 × 9 matrix, the QR decomposition of a 6 × 6 matrix and the eigendecomposition of a 4 × 4 matrix. In the second row, the implied computational complexities are summed. In the third one, the number of correspondences required for the solvers are written. The fourth row lists example outlier ratios in the data. In the fifth one, the theoretical number of iterations of RANSAC [16] is written for each outlier ratio with confidence set to 0.99. The last row shows the computational complexity, i.e. the complexity of one iteration multiplied by the number of iteration, of RANSAC combined with the minimal methods. It can be seen that the proposed method leads to significantly smaller computational complexity. Moreover, we believe that by designing a specific solver to our two quadratic equations in two unknowns, similarly as in [20], the computational complexity of our solver can be even reduced.

5.2. Synthesized tests

To test the accuracy of the homographies obtained by the proposed method, first, we created a synthetic scene consisting of two cameras represented by their 3 × 4 projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$. They were located randomly on a center-aligned sphere. A plane with random normal was generated in the origin and ten random points, lying on the plane, were projected into both cameras. The points were at most one unit far from the origin. To get the ground truth affine transformations, we calculated homography $\mathbf{H}$ by projecting four random points from the plane to the cameras and applying the normalized DLT [16] algorithm. The local affine transformation of each correspondence was computed from the ground truth homography by (1). Note that $\mathbf{H}$ could have been calculated directly from the plane parameters. However, using four points promised an indirect but geometrically interpretable way of noising the affine parameters: adding noise to the coordinates of the four projected points. To simulate the SIFT orientations and scales, $\mathbf{A}$ was decomposed to $\mathbf{J}_1$, $\mathbf{J}_2$. Since the decomposition is ambiguous, α_1, $q_{u,1}$, $q_{v,1}$, w_1 were set to random values. $\mathbf{J}_1$ was calculated from them. Finally, $\mathbf{J}_2$ was calculated as $\mathbf{J}_2 = \mathbf{A}\mathbf{J}_1$. Zero-mean Gaussian-noise was added to the point coordinates, and, also, to the coordinates which were used to estimate the affine transformations.

Fig. 3 reports the numerical stability of the methods in the noise-free case. The frequencies (vertical axis), i.e. the number of occurrences in 100 000 runs, are plotted as the function of the $\log_{10}$ average transfer error (in px; horizontal) computed from the estimated homography and the not used correspondences. It can be seen that all tested solvers are numerically stable. Fig. 4 plots the $||\mathbf{H}_{\text{est}} - \mathbf{H}_{\text{gt}}||_{\text{F}}$ errors as the function of image noise level σ (vertical axis) and the ratio (horizontal) of the camera distance, i.e. the radius of the sphere on which the cameras lie, and the object size. The homographies were normalized. The proposed 2SIFT algorithm (left) is less sensitive to the choice of both parameters than the 3ORI (middle) and 4PT (right) methods.

Fig. 5 reports the re-projection error (vertical; in pixels) as the function of the image noise σ with additional noise added to the SIFT orientations (left) and scales (right) besides the noise coming from the noisy affine transformations. In the top row, the error is plotted as the function of the image noise σ. The curves show the results on different noise levels in the orientations and scales. In the bottom

	2SIFT				3ORI [3]				4PT [16]			
steps 1 iter	6×9 SVD + 6×6 QR + 4×4 EIG $6 * 9^2 + 6^3 + 4^3 = 766$				6×9 SVD $6 * 9^2 = 486$				8×9 SVD $8 * 9^2 = 649$			
m	2				3				4			
1 - μ	0.25	0.50	0.75	0.90	0.25	0.50	0.75	0.90	0.25	0.50	0.75	0.90
# iters # comps	6 4 596	16 12 256	71 54 386	458 350 828	8 3 888	34 16 524	292 141 912	4603 2 237 058	12 7 788	71 46 079	1177 763 873	46 049 29 885 801

Table 1: The theoretical computational complexity of the solvers. The operations in the solvers (1st row – steps), the computational complexity of one estimation (2nd – 1 iter), the correspondence number required for the estimation (3rd – m), possible outlier ratios (4th – $1 - \mu$), the iteration number required for RANSAC with the confidence set to 0.95 (5th – # iters), and computation complexity of the full procedure (6th – # comps).

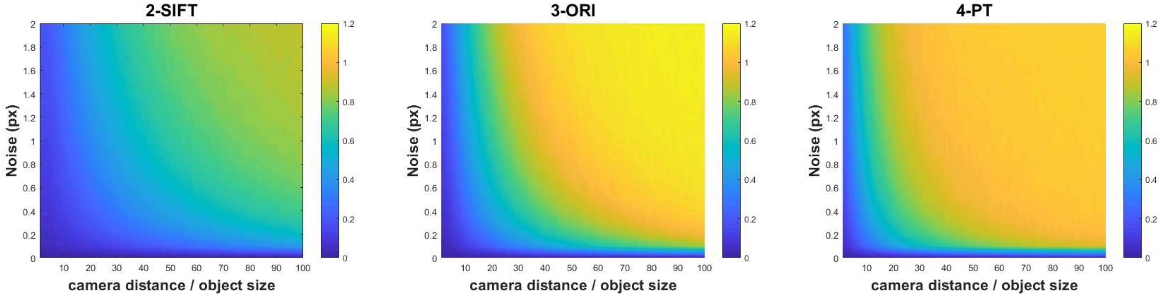


Figure 4: The average (of 10 000 runs on each noise σ) re-projection error of homography fitting to synthesized data by the proposed (2SIFT), normalized 4PT [16] and 3ORI [3] methods. Each camera is located randomly on a center-aligned sphere. Ten points from the object are projected into the cameras, and zero-mean Gaussian-noise is added to the coordinates. The affine parameters are calculated from the noisy coordinates. The re-projection error (in px; shown by color) is plotted as the function of the ”camera distance from the object / object size” ratio (horizontal) and the noise σ (in px; vertical).

row, the error is plotted as the function of the orientation (left plot) and scale (right) noise. The noise in the point coordinates was set to 1.0 px. The scale noise for the left plot was set to 1%. The orientation noise for the right one was set to 1°. It can be seen that, even for large noise in the scale and orientation, the new solver performs reasonably well.

5.3. Real world tests

To test the proposed method on real-world data, we downloaded the AdelaideRMF[1], Multi-H[2], Malaga[3] and Strecha[4] datasets. AdelaideRMF and Multi-H consist of image pairs of resolution from 455×341 to 2592×1944 and manually annotated (assigned to a homography or to the outlier class) correspondences. Since the reference point sets do not contain rotations and scales, we detected points applying the SIFT detector. The correspondences provided in the datasets were used to estimate ground truth homographies. For each homography, we selected the points out of the detected SIFT correspondences which are closer than a manually set inlier-outlier threshold, i.e. 2 pixels. As robust estimator, we chose GC-RANSAC [7] since it is state-of-the-art and its implementation is available[5]. GC-RANSAC is a locally optimized RANSAC with PROSAC [12] sampling. For fitting to a minimal sample, GC-RANSAC used one of the compared methods, e.g. the proposed one. For fitting to a non-minimal sample, the normalized 4PT algorithm was applied.

Given an image pair, the procedure to evaluate the estimators on AdelaideRMF and Multi-H is as follows: first, the ground truth homographies, estimated from the manually annotated correspondence sets, were selected one by one. For each homography: (i) The correspondences which did not belong to the selected homography were replaced by completely random correspondences to reduce the probability of finding a different plane than what was currently tested. (ii) GC-RANSAC was applied to the point set consisting of the inliers of the homography and outliers. (iii) The estimated homography is compared to the ground truth one estimated from the manually selected inliers.

The Strecha dataset consists of image sequences of buildings. All images are of size 3072×2048. The methods were applied to all possible image pairs in each sequence. The Malaga dataset was gathered entirely in urban scenarios with a car equipped with several sensors, including a

[1] cs.adelaide.edu.au/~hwong/doku.php?id=data
[2] web.eee.sztaki.hu/~dbarath
[3] www.mrpt.org/MalagaUrbanDataset
[4] https://cvlab.epfl.ch/

[5] https://github.com/danini/graph-cut-ransac

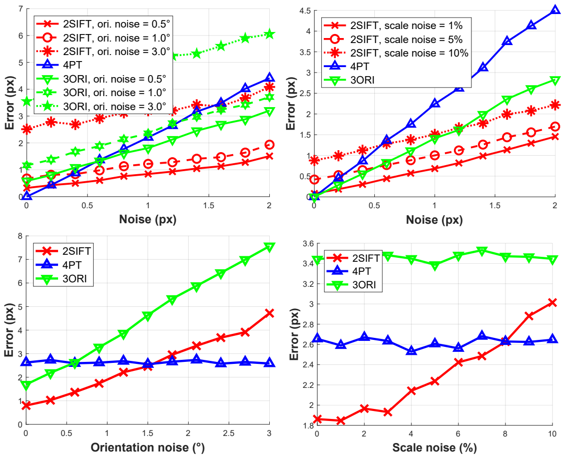


Figure 5: The average (10 000 runs on each noise σ) re-projection error of homography fitting to synthesized data by the 2SIFT, normalized 4PT [16] and 3ORI [3] methods. The same test scene is used as in Figure 4. For each plot, additional noise was added to the orientations or the scales besides the noise coming from the noisy affine transformations. (*Top*) The error is plotted as the function of the image noise σ. The curves show the results on different noise levels in the orientations and scales. (*Bottom*) The error is plotted as the function of the orientation (left plot) and scale (right) noise. The noise in the point coordinates was set to 1.0 px. The scale noise for the left plot was set to 1%. The orientation noise for the right one was set to 1°.

		2SIFT	3ORI [3]	4PT [16]
AdelaideRMF (43#)	ϵ (px)	**1.57**	1.97	1.61
	# iters.	**877**	9 772	26 082
	time (s)	**0.092**	0.918	2.989
Multi-H (33#)	ϵ (px)	1.90	3.41	**1.87**
	# iters.	**80 031**	458 800	410 781
	time (s)	**57.921**	213.900	300.645
Strecha (852#)	ϵ (px)	1.42	1.51	**1.25**
	# iters.	**4 718**	17 414	60 973
	time (s)	**1.435**	3.180	10.246

Table 2: Homography estimation on the `AdelaideRMF` (18 pairs; 43 planes) and `Multi-H` (4 pairs; 33 planes) and `Strecha` datasets (852 planes) by GC-RANSAC [7] combined with minimal methods. Each column reports the results of a method. The required confidence was set to 0.95. The reported properties are the mean re-projection error (ϵ, in pixels); the number of samples drawn by GC-RANSAC (# iters.); and the processing time in seconds. Average of 100 runs on each image pair.

high-resolution camera and five laser scanners. 15 video sequences are provided and we used every 10th image from each sequence. The ground truth projection matrices are provided for both datasets. To get a reference correspondence set for each image pair in the `Strecha` and `Malaga` datasets, first, calculated the fundamental matrix from the ground truth camera poses provided in the datasets. SIFT detector was applied. Correspondences were selected for which the symmetric epipolar distance was smaller than 1.0 pixel. RANSAC was applied to the filtered correspondences finding the most dominant homography with a threshold set to 1.0 pixel and confidence to 0.9999. The inliers of this homography were considered as a reference set. In case of having less then 50 reference points, the pair was discarded from the evaluation. In total, 852 image pairs were tested in the `Strecha` dataset and 9 064 pairs in the `Malaga` dataset.

Example results are shown in Fig. 2. The inliers of the homography estimated by 2SIFT are drawn. Also, the number of iteration required for 2SIFT and 4PT and the ground truth inlier ratios are reported. In all cases, *2SIFT made significantly fewer iterations* than 4PT.

Table 2 reports the results on the `AdelaideRMF` (rows 2–4), `Multi-H` (5–7) and `Strecha` (8–10) datasets. The names of the datasets are written into the first column and the numbers of planes are in brackets. The names of the tested techniques are written in the first row. Each block, consisting of three rows, shows the mean re-projection error computed from the manually annotated correspondences and the estimated homographies (ϵ; in pixels; avg. of 100 runs on each pair); the number of samples drawn by the outer loop of GC-RANSAC (# iters.); and the processing time (in secs). The RANSAC confidence was set to 0.95 and the inlier-outlier threshold to 2 pixels. It can be seen that the proposed method has similar errors to that of the 4PT algorithm, but *2SIFT leads to 1–2 orders of magnitude speedup* compared to 4PT.

The results on the `Malaga` dataset are shown in Figure 6. The confidence of GC-RANSAC was set to 0.95 and the inlier-outlier threshold to 2.0 pixels. The reported properties are the average re-projection error (left; in pixels), processing time (middle; in seconds) and the average number of iterations (right). It can be seen that the re-projection errors of 4PT and 2SIFT are fairly similar However, *2SIFT is significantly faster in all cases* due to making much fewer iterations than 4PT.

6. Conclusion

We proposed a theoretically justifiable interpretation of the angles and scales which the orientation- and scale-covariant feature detectors, e.g. SIFT or SURF, provide. Building on this, two new general constraints are proposed for covariant features. These constraints are then exploited to derive two new formulas for homography estimation. Using the derived equations, a solver is proposed for estimating the homography from two correspondences. The new solver is numerically stable and easy to implement. More-

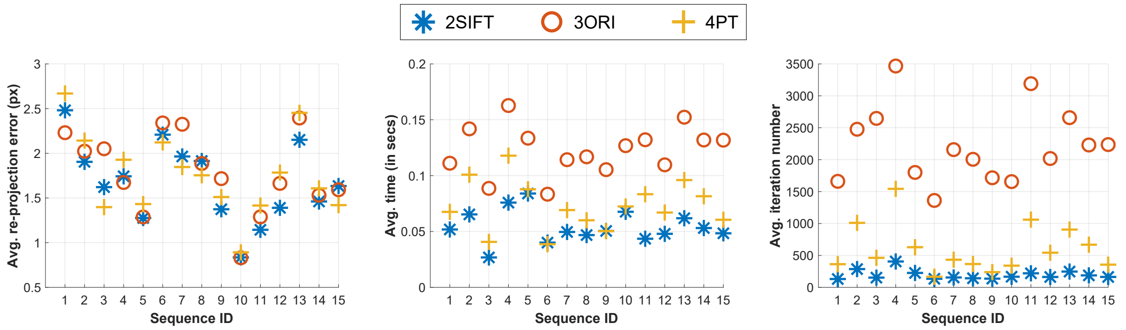


Figure 6: The results on 15 sequences (9 064 image pairs) of the `Malaga` dataset using GC-RANSAC [7] as a robust estimator and different minimal solvers (2SIFT, 3ORI, 4PT). The confidence of RANSAC was set to 0.95 and the inlier-outlier threshold to 2.0 pixels. The re-projection error (left; in pixels), average processing time (middle; in seconds) and average iteration number (right) are reported.

over, it leads to results superior in terms of geometric accuracy in many cases. Also, it is shown how the normalization of the point correspondences affects the rotation and scale parameters. Due to requiring merely two feature pairs, robust estimators, e.g. RANSAC, do significantly fewer iterations than by using the four-point algorithm. The method is tested in a synthetic environment and on publicly available real-world datasets consisting of thousands of image pairs. The source code is available at `https://github.com/danini/homography-from-sift-features`.

Acknowledgement

Z. Kukelova was supported by the ESI Fund, OP RDE program under project International Mobility of Researchers MSCA-IF at CTU No. CZ.02.2.69/0.0/0.0/17_050/0008025. D. Barath was supported by the Hungarian Scientific Research Fund (No. NKFIH OTKA KH-126513 and K-120499) and by the OP VVV project CZ.02.1.01/0.0/0.0/16019/000076 Research Center for Informatics.

A. Proof the affine decomposition

We prove that decomposition $\mathbf{A} = \mathbf{J}_2\mathbf{J}_1^{-1}$, where $\mathbf{J}_i$ is the Jacobian of the projection function w.r.t. the directions in the ith image, is geometrically valid. Suppose that a three-dimensional point $\mathbf{P} = \begin{bmatrix} x & y & z \end{bmatrix}^\text{T}$ lying on a continuous surface S is given. Its projection in the ith image is $\mathbf{p}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\text{T}$. The projected coordinates, u_i and v_i, are determined by the projection functions $\mathbf{\Pi}_u, \mathbf{\Pi}_v : \mathbb{R}^3 \to \mathbb{R}$ as follows: $u_i = \mathbf{\Pi}_u^i(x,y,z)$, $v_i = \mathbf{\Pi}_v^i(x,y,z)$, where the coordinates of the surface point are written in parametric form as $x = \mathcal{X}(u,v)$, $y = \mathcal{Y}(u,v)$, $z = \mathcal{Z}(u,v)$. It is well-known in differential geometry [18] that the basis of the tangent plane at point $\mathbf{P}$ is written by the partial derivatives of S w.r.t. the spatial coordinates. The surface normal $\mathbf{n}$ is expressed by the cross product of the tangent vectors $\mathbf{s}_u$ and $\mathbf{s}_v$ where $\mathbf{s}_u = \begin{bmatrix} \frac{\partial \mathcal{X}(u,v)}{\partial u} & \frac{\partial \mathcal{Y}(u,v)}{\partial u} & \frac{\partial \mathcal{Z}(u,v)}{\partial u} \end{bmatrix}^\text{T}$, and $\mathbf{s}_v$ is calculated similarly. Finally, $\mathbf{n} = \mathbf{s}_u \times \mathbf{s}_v$. Locally, around point $\mathbf{P}$, the surface can be approximated by the tangent plane, therefore, the neighboring points in the ith image are written as the first-order Taylor-series as follows:

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

where $[\Delta v, \Delta u]^\text{T}$ is the translation on surface S, and Δx, Δy are the coordinates of the implied translation added to $\mathbf{p}_i$. It can be seen that transformation $\mathbf{J}_i$ mapping the infinitely close vicinity around point $\mathbf{p}_i$ in the ith image is given as $\mathbf{J}_i = \begin{bmatrix} \frac{\partial \Pi_x^i(x,y,z)}{\partial u} & \frac{\partial \Pi_x^i(x,y,z)}{\partial v} \\ \frac{\partial \Pi_y^i(x,y,z)}{\partial u} & \frac{\partial \Pi_y^i(x,y,z)}{\partial v} \end{bmatrix}$, thus $\begin{bmatrix} \Delta x & \Delta y \end{bmatrix}^\text{T} \approx \mathbf{J}_i \begin{bmatrix} \Delta u & \Delta v \end{bmatrix}^\text{T}$. The partial derivatives are reformulated using the chain rule. As an example, the first element it is as

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

where $\nabla\mathbf{\Pi}_x^i$ is the gradient vector of $\mathbf{\Pi}_x$ w.r.t. coordinates x, y and z. Similarly,

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

Therefore, $\mathbf{J}_i$ can be written as $\mathbf{J}_i = \begin{bmatrix} \nabla(\mathbf{\Pi}_x^i)^\text{T} \\ \nabla(\mathbf{\Pi}_y^i)^\text{T} \end{bmatrix} \begin{bmatrix} \mathbf{s}_u & \mathbf{s}_v \end{bmatrix}$. Local affine transformation $\mathbf{A}$ transforming the infinitely close vicinity of point $\mathbf{p}_1$ in the first image to that of $\mathbf{p}_2$ in the second one is as follows:

$$\begin{bmatrix} \Delta x_2 \\ \Delta y_2 \end{bmatrix} = \mathbf{J}_2\mathbf{J}_1^{-1} \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} \Delta x_1 \\ \Delta y_1 \end{bmatrix}.$$

References

[1] Daniel Barath. P-HAF: Homography estimation using partial local affine frames. In *International Conference on Computer Vision Theory and Applications*, 2017. 2

[2] Daniel Barath. Approximate epipolar geometry from six rotation invariant correspondences. In *International Conference on Computer Vision Theory and Applications*, 2018. 2

[3] Daniel Barath. Five-point fundamental matrix estimation for uncalibrated cameras. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 5, 6, 7

[4] Daniel Barath. Recovering affine features from orientation- and scale-invariant ones. In *Asian Conference on Computer Vision*, 2018. 2, 3

[5] Daniel Barath and Levente Hajder. A theory of point-wise homography estimation. *Pattern Recognition Letters*, 94:7–14, 2017. 1

[6] Daniel Barath and Levente Hajder. Efficient recovery of essential matrix from two affine correspondences. *IEEE Transactions on Image Processing*, 27(11):5328–5337, 2018. 1, 4

[7] Daniel Barath and Jiri Matas. Graph-Cut RANSAC. In *Conference on Computer Vision and Pattern Recognition*, 2018. 2, 4, 6, 7, 8

[8] Daniel Barath, J. Molnár, and Levente Hajder. Optimal surface normal from affine transformation. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SciTePress, 2015. 1, 3

[9] Daniel Barath, T. Toth, and Levente Hajder. A minimal solution for two-view focal-length estimation using two affine correspondences. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *European Conference on Computer Vision*, 2006. 1, 2

[11] Jacob Bentolila and Joseph M. Francos. Conic epipolar constraints from affine correspondences. *Computer Vision and Image Understanding*, 2014. 1

[12] Ondrej Chum and Jiri Matas. Matching with PROSAC-progressive sample consensus. In *Computer Vision and Pattern Recognition*, 2005. 6

[13] David Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*. Springer-Verlag New York, 2nd edition, 2005. 3, 4

[14] Daniel Grayson and Michael Stillman. Macaulay2, a software system for research in algebraic geometry. available at `www.math.uiuc.edu/Macaulay2/`. 3

[15] Richard Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence*, 1997. 3, 4

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 2, 5, 6, 7

[17] Kevin Köser. *Geometric Estimation with Local Affine Frames and Free-form Surfaces*. Shaker, 2009. 1

[18] Erwin Kreyszig. *Introduction to differential geometry and Riemannian geometry*, volume 16. University of Toronto Press, 1968. 8

[19] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conference on Computer Vision*, volume 5304 of *Lecture Notes in Computer Science*, 2008. 4

[20] Zuzana Kukelova, Jan Heller, and Andrew Fitzgibbon. Efficient intersection of three quadrics and applications in computer vision. In *Conference on Computer Vision and Pattern Recognition*, pages 1799–1808, 2016. 5

[21] Zuzana Kukelova, Joe Kileel, Bernd Sturmfels, and Tomas Pajdla. A clever elimination strategy for efficient minimal solvers. In *Conference on Computer Vision and Pattern Recognition*, 2017. http://arxiv.org/abs/1703.05289. 3

[22] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer vision*, 1999. 1, 2

[23] Jiri Matas, Ondrej Chum, Martin Urban, and Tomas Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004. 2

[24] Kristian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005. 1

[25] Steven Mills. Four-and seven-point relative camera pose from oriented features. In *International Conference on 3D Vision*, pages 218–227. IEEE, 2018. 2

[26] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 2015. 1, 2

[27] J. Molnár and D. Chetverikov. Quadratic transformation for planar mapping of implicit surfaces. *Journal of Mathematical Imaging and Vision*, 2014. 2

[28] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009. 1

[29] Michal Perdoch, Jiri Matas, and Ondrej Chum. Epipolar geometry from two correspondences. In *International Conference on Pattern Recognition*, 2006. 1

[30] James Pritts, Zuzana Kukelova, Viktor Larsson, and Ondrej Chum. Radially-distorted conjugate translations. *Conference on Computer Vision and Pattern Recognition*, 2018. 1

[31] Carolina Raposo and Joao P. Barreto. πmatch: Monocular vslam and piecewise planar reconstruction using fast plane correspondences. In *European Conference on Computer Vision*, pages 380–395. Springer, 2016. 1

[32] Carolina Raposo and Joao P. Barreto. Theory and practice of structure-from-motion using affine correspondences. In *Computer Vision and Pattern Recognition*, 2016. 1

[33] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to sift or surf. In *International Conference on Computer Vision*, pages 2564–2571. IEEE, 2011. 2