

Quantitative Affine Feature Detector Comparison based on Real-World Images Taken by a Quadcopter

Zoltán Pusztai^{1,2} and Levente Hajder²

¹*Geometric Computer Vision Group, Machine Perception Laboratory, MTA SZTAKI, Kende st. 17, Budapest 1111, Hungary*

²*Department of Algorithm and Applications, Eötvös Loránd University, Pázmány Péter stny. 1/C, Budapest 1117, Hungary*

Keywords: Feature detector, Quantitative Comparison, Affine Transformation, Detection Error, Ground Truth Generation.

Abstract: Feature detectors are frequently used in computer vision. Recently, detectors which can extract the affine transformation between the features have become popular. With affine transformations, it is possible to estimate the properties of the camera motion and the 3D scene from significantly fewer feature correspondences. This paper quantitatively compares the affine feature detectors on real-world images captured by a quadcopter. The ground truth (GT) data are calculated from the constrained motion of the cameras. Accurate and very realistic testing data are generated for both the feature locations and the corresponding affine transformations. Based on the generated GT data, many popular affine feature detectors are quantitatively compared.

1 INTRODUCTION

Feature detectors have been studied since the born of computer vision. First, point-related local detectors are developed to explore properties of camera motion and epipolar geometry. Local features describe only a small area of the image, thus they can be effectively used to find point correspondences, even in the presence of high illumination, viewpoint change or occlusion. Affine feature detectors can detect the local affine warp of the detected point regions. The affine transformation can be used to solve the basic task of epipolar geometry (e.g. to detect camera motion, object detection or 3D reconstruction) using less correspondences than general point features. This paper deals with the quantitative comparison of affine feature detectors using video sequences, taken by a quadcopter, captured in a real-world environment.

Interest feature detectors have been studied in a long period of computer vision. The well-known Harris (Harris and Stephens, 1988) corner detector or Shi-Tomasi detector (Shi and Tomasi, 1994) have been published more than two decades ago. Since then, new point feature detectors have been implemented, e.g. SIFT (Lowe, 2004), SURF (Bay et al., 2008), KAZE (Alcantarilla et al., 2012), BRISK (Leutenegger et al., 2011) and so on. Correspondences are made using a feature location and a feature descriptor. The latter describes the local small area of the feature with

a vector in a compact and distinguish way. These descriptor vectors can be used for feature matching over the successive image. Features, whose descriptor vectors are close to each other, potentially yield a match.

While point-based features estimate only point correspondences, affine feature detectors can extract the affine transformations around the feature centers as well. An affine transformation contains the linear approximation of the warp around the point correspondences. In other words, this is a 2 by 2 linear transformation matrix which transforms the local region of the feature to that of the corresponding feature. Each affine transformation contains enough information for estimating the normal vector of the tangent plane at the corresponding 3D location of the feature. These additional constraints can be used to estimate fundamental matrix, camera movement or other epipolar properties using less number of features, than using point correspondences.

The aim of this paper is to quantitatively compare the feature detectors and the estimated affine transformations, using real-world video sequences captured by a quadcopter. The literature of previous work is not rich. Maybe the most significant work was published by Mikolajczyk et al. (Mikolajczyk et al., 2005). They compare several affine feature detectors using real-world images. However, in their comparison, either the camera has fixed location or the scenes are planar, thus, the images are related by homographies. The er-

ror for the affine transformations is computed by the overlapping error of the related affine ellipses or by the repeatability score. Even though the authors compared the detectors using several different noise types (blur, JPEG compression, light change) we think that the constraints of non-moving camera or planar scene yield very limited test cases. A comprehensive study can be found in (Tuytelaars and Mikolajczyk, 2008), however the paper does not contain any real-world test. Recently Pusztai et al. proposed a technique to quantitatively compare feature detectors and descriptors using a structured-light 3D scanner (Pusztai and Hajder, 2017). However, the testing data consist of small objects and rotation movement only. Tareen et al. (Tareen and Saleem, 2018) also published a comparison of the most popular feature detector and descriptor algorithms. Their ground-truth (GT) data generation is twofold: (i) They use the Oxfordian dataset, that was also applied in the work of (Mikolajczyk et al., 2005), (ii) they generated test images, including GT transformations, carried out by different kinds of affine transformations for the original images: translation, rotation and scale, and the transformed images are synthesized by common bilinear interpolation. In the latter case, the processed images are not taken by a camera, thus the input data for the comparison is not really realistic.

The literature of affine feature comparisons is not extensive, despite the fact that it is an important subject. Most detectors are compared to others, using only a small set of images, and parameters tuned to achieve the best results. In real world applications, the best parameter set may differ from the laboratory experience. Thus, more comparisons have to be made using real-world video sequences and various camera movements.

In this paper, we show, that the affine invariant feature detectors can be evaluated quantitatively on real-world test sequences if images are captured by a quadcopter-mounted camera. The main contributions of the paper are twofold: (i) First, the GT affine transformation generation is shown in case of several special movements of a quadcopter where affine transformations and corresponding point locations can be very accurately determined. To the best of our knowledge, this is the first study in which the ground truth data is generated using real images of a moving cop-ter. (ii) Then several affine covariant feature detectors are quantitatively compared using the generated GT data. Both point locations and the related affine transformations are examined in the comparison.

The structure of this paper is as follows. First, the rival methods are theoretically described in Section 2. Then the ground truth data generation methods are

overviewed for different drone movements and camera orientations. Section 4 contains the methodology of the evaluation. The test results are discussed in Section 5, and Section 6 concludes the research.

2 OVERVIEW OF AFFINE-COVARIANT DETECTORS

In this section, the affine transformations and the affine covariant detectors are briefly introduced. The detectors aim to separately find discriminate features in the images. If a feature is found, then the affine shape can be determined which is usually visualized as an ellipse. Figure 1 shows the local affine regions of a corresponding feature pair in successive images. The methods to detect discriminate features and their affine shapes vary from detector to detector. They are briefly introduced as follows:

Harris-Laplace, Harris-Affine. The methods introduced by (Mikolajczyk and Schmid, 2002) are based on the well-known Harris detector (Harris and Stephens, 1988). Harris uses the so-called second moment matrix to extract features in the images. The matrix is as follows:

$$M(\mathbf{x}) = \sigma_D^2 G(\sigma_I) * \begin{bmatrix} f_x^2(\mathbf{x}, \sigma_D) & f_x(\mathbf{x}, \sigma_D) f_y(\mathbf{x}, \sigma_D) \\ f_x(\mathbf{x}, \sigma_D) f_y(\mathbf{x}, \sigma_D) & f_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}, \quad (1)$$

where

$$G(\sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|\mathbf{x}|^2}{2\sigma^2}\right),$$

$$f_x(\mathbf{x}, \sigma_D) = \frac{\partial}{\partial \mathbf{x}} G(\sigma_D) * f(\mathbf{x}).$$

The matrix (M) contains the gradient distribution around the feature, σ_D , σ_I are called the differentiation scale and integration scale, respectively. A local feature point is found, if the term $\det(M) - \lambda \text{trace}(M)$ is higher than a pre-selected threshold. This means that both of the eigenvalues of M are large, which indicates a corner in the image.

After the location of the feature is found, a characteristic scale selection needs to be carried out. The circular Laplace operator is used for this purpose. The characteristic scale is found if the similarity of the operator and the underlying image structure is the highest. The final step of this affine invariant detector is to determine the second scale of the feature points using the following iterative estimation:

1. Detect the initial point and corresponding scale.

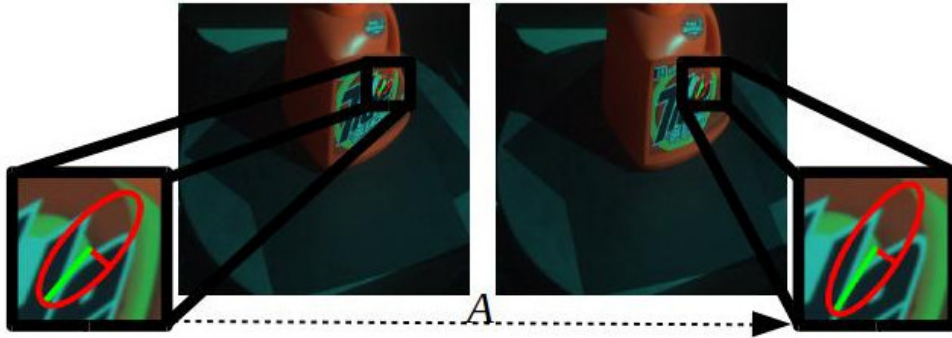


Figure 1: The affine transformation (A) of corresponding features. The ellipses show the affine shapes around the features. A approximately transforms the local region of the first image to that of the second one.

2. Estimate the affine shape using M .
 3. Normalize the affine shape into a circle using $M^{1/2}$.
 4. Detect new position and scale in the normalized region.
 5. Goto (2), if the the eigenvalues of M are not equal.
- The iteration always converges, the obtained affine shape is described as an ellipse.

The scale and shape selections described above can be applied to any point feature. Mikolajczyk also proposed the **Hessian-Laplace** and **Hessian-Affine** detectors, which use the Hesse matrix for extracting features, instead of the Harris. The matrix is as follows:

$$H(\mathbf{x}) = \begin{bmatrix} f_{xx}(\mathbf{x}, \sigma_D) & f_{xy}(\mathbf{x}, \sigma_D) \\ f_{xy}(\mathbf{x}, \sigma_D) & f_{yy}(\mathbf{x}, \sigma_D) \end{bmatrix}, \quad (2)$$

where $f_{xx}(\mathbf{x}, \sigma_D)$ is the second order Gaussian smoothed image derivatives. The Hesse matrix can be used to detect blob-like features.

Edge-based Regions (EBR). (Tuytelaars and Van Gool, 2004) introduced a method to detect affine covariant regions around the corners. The Harris corner detector is used along with standard Canny edge detector. Affine regions are found where two edges meet at a corner. The corner point (\mathbf{p}) and two points moving along the two edges (\mathbf{p}_1 and \mathbf{p}_2) define a parallelogram. The final shape is found, where the region yields extremum in the following function:

$$f(\Omega) = abs \left(\frac{|(\mathbf{p} - \mathbf{p}_g)(\mathbf{q} - \mathbf{p}_g)|}{|(\mathbf{p} - \mathbf{p}_1)(\mathbf{p} - \mathbf{p}_2)|} \right) \times \frac{M_{00}^1}{\sqrt{M_{00}^2 M_{00}^0 - (M_{00}^1)^2}}, \quad (3)$$

where

$$M_{pq}^n = \int_{\Omega} I^n(x, y) x^p y^q dx dy,$$

$$\mathbf{p}_g = \left(\frac{M_{00}^1}{M_{00}^0}, \frac{M_{01}^1}{M_{00}^0} \right),$$

and \mathbf{p}_g is the center of gravity. The parallelogram regions are then converted to ellipses.

Intensity-extrema-based Regions (IBR). While EBR find features at the corners and edges, IBR extracts affine regions based on intensity properties. First, the image is smoothed, then the local extrema is selected using non-maximum suppression. These points cannot be detected precisely, however they are robust to monotonic intensity transformations. Rays are cast from the local extremum to every direction, and the following function is evaluated on each ray:

$$f_I(t) = \frac{abs(I(t) - I_0)}{\max \left(\frac{\int_0^t abs(I(t) - I_0) dt}{t}, d \right)} \quad (4)$$

where, t , $I(t)$, I_0 and d are the arclength along the ray, the intensity at position, the intensity extremum and a small number to prevent dividing by 0, respectively. The function yields extremum where the intensity suddenly changes. The points along the cast rays define an usually irregularly-shaped region, which is replaced by an ellipse having the same moments up to the second order.

TBMR. Tree-Based Morse Regions is introduced in (Xu et al., 2014). This detector is motivated by Morse theory, selecting critical regions as features, using the Min and Max-tree. TBMR can be seen as a variant of MSER, however, TBMR is invariant to illumination change and needs less number of parameters.

SURF. SURF is introduced in (Bay et al., 2008). It is the fast approximation of SIFT (Lowe, 2004). The Hessian matrix is roughly approximated using box filters, instead of Gaussian filters. This makes the detector relatively fast comparing to the others. Despite the approximations, SURF can find reliable features.

3 GT DATA GENERATION¹

In order to quantitatively compare the detectors performance and the reliability of affine transformations, ground truth (GT) data is needed. Many comparison databases (Mikolajczyk et al., 2005; Cordes et al., 2013; Zitnick and Ramnath, 2011) are based on homography, which is extracted from the observed planar object. Images are taken from different camera positions, and the GT affine transformations are calculated from the homography. Instead of this, our comparison is based on images captured by a quadcopter in a real world environment and the GT affine transformation is extracted directly from constrained movement of the quadcopter. This work is motivated by the fact, that the affine parameters can be calculated more precisely from the constrained movements, than from the homography. Moreover, if the parameters of the motion are estimated, the GT location of the features can be determined as well. Thus, it is possible to compare not only the affine transformations, but the locations of the features, additionally. In this section, these restricted movements, and the computations of the corresponding affine transformations are introduced.

3.1 Rotation

In case of the rotation movement, the quadcopter stays at the same position, and rotates around its vertical axis. Thus, the translation vector of the movement is equivalent to $\mathbf{0}$ all the time. The rotation matrices between the images can be computed if the degrees and center of the rotation are known. Example images are given in Fig. 2. The first row shows the images taken by the quadcopter, and the second row shows colored boxes, which are related by affine transformations.

Let α_i be the degree of rotation in radians, then the rotation matrix is defined as follows:

$$\mathbf{R}_i = \begin{bmatrix} \cos \alpha_i & -\sin \alpha_i \\ \sin \alpha_i & \cos \alpha_i \end{bmatrix}. \quad (5)$$

This matrix describes the transformation of corresponding affine shapes, in case of the rotation movement. Let us assume that corresponding feature points in the images are given, then the relation of the corresponding features can be expressed as follows:

$$\mathbf{p}_i^f = \mathbf{v} + \mathbf{R}_f (\mathbf{p}_i^1 - \mathbf{v}), \quad (6)$$

where \mathbf{p}_p^f , α_i , \mathbf{v} are the p -th feature location in the f -th image, the degree of rotation between the first

to the f -th image and the center of rotation, respectively. The latter one is considered constant during the rotation movement.

To estimate the degree and center of the rotation, the Euclidean distances of the selected and estimated features have to be minimized. The cost function describing the error of estimation is as follows:

$$\sum_{f=2}^F \sum_{i=1}^P \left\| \mathbf{p}_i^f - \mathbf{R}_f (\mathbf{p}_i^1 - \mathbf{v}) - \mathbf{v} \right\|_2^2, \quad (7)$$

where F and P are the number of frames and selected features, respectively. The minimization can be solved by an alternation algorithm. The alternation itself consist of two steps: (i) estimation of the center of rotation \mathbf{v} and (ii) estimation of rotation angles α_i .

3.1.1 Estimation of Rotation Center

The problem of rotation center estimation can be formalized as $\mathbf{A}\mathbf{v} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}_2 - \mathbf{I} \\ \vdots \\ \mathbf{R}_F - \mathbf{I} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{R}_2 \mathbf{p}_1^1 - \mathbf{p}_1^2 \\ \vdots \\ \mathbf{R}_F \mathbf{p}_P^1 - \mathbf{p}_P^F \end{bmatrix}. \quad (8)$$

The optimal solution in the least-squares sense is given by the pseudo-inverse of \mathbf{A} :

$$\mathbf{v} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (9)$$

3.1.2 Estimation of Rotating Angles

The rotation angles are separately estimated for each image. The estimation can be written in a linear form $\mathbf{C}\mathbf{x} = \mathbf{d}$ subject to $\mathbf{x}^T \mathbf{x} = 1$, where

$$\mathbf{C} \begin{bmatrix} \cos \alpha_i \\ \sin \alpha_i \end{bmatrix} = \mathbf{d}. \quad (10)$$

The coefficient matrix \mathbf{C} and vector \mathbf{d} are as follows:

$$\mathbf{C} = \begin{bmatrix} x_1^1 - u & v - y_1^1 \\ y_1^1 - v & x_1^1 - u \\ \vdots & \vdots \\ x_P^1 - u & v - y_P^1 \\ y_P^1 - v & x_P^1 - u \end{bmatrix}, \mathbf{d} = \begin{bmatrix} x_1^f - u \\ y_1^f - v \\ \vdots \\ x_P^f - u \\ y_P^f - v \end{bmatrix}, \quad (11)$$

where $[x_i^f, y_i^f]^T = \mathbf{p}_i^f$ and $[u, v]^T = \mathbf{v}$.

The optimal solution for this problem is given by one of the roots of a four degree polynomial as it is written in the appendix.

Convergence. The steps described above are repeated one after the other, iteratively. The speed of convergence does not matter for our application. However, we empirically found that it convergences after a few

¹Testing data are submitted as supplementary material.



(a) Example images from the sequence.



(b) Colored boxes indicate corresponding areas computed by the rotation parameters. Images are the same as in the first row.

Figure 2: Images are taken while the quadcopter is rotating around its vertical axis, while its position is fixed.

iterations, when the center of the image is used as the initial value for the center of the rotation.

Ground Truth Affine Transformations. The affine transformations are easy to determine: they are equal to the rotation: $\mathbf{A} = \mathbf{R}$.

3.2 Uniform Motion: Front View

For this test case, the quadcopter moves along a straight line and does not rotate. Thus, the rotation matrix relative to the camera movement is equal to the identity. The camera faces to the front, thus the Focus of Expansion (FOE) can be computed. The FOE is the projection of the spatial line of movement at the infinity to the camera image. It is also an epipole in the image, meaning that all epipolar lines intersect at the FOE. The epipolar lines can be determined from the projections of corresponding spatial points. Fig. 3 shows an example image sequence for this motion. In the second row, the red dot indicates the calculated FOE, and the blue lines mark the epipolar lines.

The maximum likelihood estimation of the FOE can be solved by a numerical minimization of the sum of squared orthogonal distances from the projected points and the measured epipolar lines (Hartley and Zisserman, 2003). Thus, the cost function to be minimized contains the sum of all feature distance to the related epipolar line. It can be formalized as follows:

$$\sum_{f=1}^F \sum_{i=1}^P \left(\left(\mathbf{p}_i^f - \mathbf{m} \right)^T \begin{bmatrix} -\sin \beta_i \\ \cos \beta_i \end{bmatrix} \right)^2, \quad (12)$$

where \mathbf{m} and β_i are the FOE and the angle between the epipolar line and the X axis. Note, that the expression $[-\sin \beta_i, \cos \beta_i]^T$ is the normal vector of the i -th epipolar line. This cost function can be minimized with an alternation, iteratively refining the FOE and angles of epipolar lines. The center of the image is used as the initial value for the FOE, then the epipolar lines can be calculated as it is described in the following subsection.

3.2.1 Estimation of Epipolar Lines

The epipolar lines intersect at the FOE, because a pure translation motion is considered. Moreover, the epipolar lines connecting corresponding features with the FOE are the same along the images. Each angle between the epipolar lines and the horizontal (image) axis can be computed as a homogeneous system of equations $\mathbf{A}\mathbf{x} = 0$ with constraint $\mathbf{x}^T \mathbf{x} = 1$ as follows:

$$\mathbf{A} = \begin{bmatrix} y_i^1 - m_y & m_x - x_i^1 \\ \vdots & \vdots \\ y_i^F - m_y & m_x - x_i^F \end{bmatrix}, \mathbf{x} = \begin{bmatrix} \cos \beta_i \\ \sin \beta_i \end{bmatrix}, \quad (13)$$

where $[m_x, m_y] = \mathbf{m}$ and $[x_i^f, y_i^f] = \mathbf{p}_i^f$ are the coordinates of the FOE and that of the selected features, respectively.

The solution which minimizes the cost function is obtained as the eigenvector (\mathbf{v}) of the smallest eigenvalue of matrix $\mathbf{A}^T \mathbf{A}$. Then, $\beta_i = \text{atan2}(v_y, v_x)$.

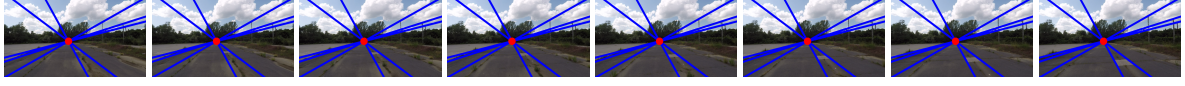
3.2.2 Estimation of the Focus of Expansion

The FOE is located where the epipolar lines of the features intersect, thus the estimation of the FOE can be formalized as a linear system of equations $\mathbf{A}\mathbf{m} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} -\sin \beta_1 & \cos \beta_1 \\ -\sin \beta_1 & \cos \beta_1 \\ \vdots & \vdots \\ -\sin \beta_1 & \cos \beta_1 \\ -\sin \beta_2 & \cos \beta_2 \\ \vdots & \vdots \\ \vdots & \vdots \\ -\sin \beta_N & \cos \beta_N \end{bmatrix}, \quad (14)$$



(a) A few images of the sequence.



(b) Red dot marks the FOE, blue lines are the epipolar ones. Images are the same as in the first row.

Figure 3: The images are taken while the quadcopter is moving parallel to the ground.

and

$$\mathbf{b} = \begin{bmatrix} y_1^1 * \cos \beta_1 - x_1^1 \sin \beta_1 \\ y_1^2 * \cos \beta_1 - x_1^2 \sin \beta_1 \\ \vdots \\ y_1^F * \cos \beta_1 - x_1^F \sin \beta_1 \\ y_2^F * \cos \beta_2 - x_2^F \sin \beta_2 \\ \vdots \\ y_N^F * \cos \beta_N - x_N^F \sin \beta_N \end{bmatrix}. \quad (15)$$

This system of equations can be solved with the Pseudo inverse of \mathbf{A} . Thus $\mathbf{v} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A} \mathbf{b}$.

3.2.3 The Fundamental Matrix

The steps explained above are iteratively repeated until convergence. The GT affine transformation is impossible to compute in an unknown environment, because it depends on the surface normal (Barath et al., 2015), and the normals are varying from feature to feature. However, some constraints can be achieved if the fundamental matrix is known. The fundamental matrix describes the transformation of epipolar lines between stereo images in a static environment. It is the composition of the camera matrices and the parameters of camera motion as follows:

$$\mathbf{F} = \mathbf{K}^{-T} \mathbf{R} [\mathbf{t}]_x \mathbf{K}^{-1}, \quad (16)$$

where \mathbf{K} , \mathbf{R} and $[\mathbf{t}]_x$ are the camera matrix, rotation matrix and the matrix representation of the cross product, respectively. The fundamental matrix can be computed as $\mathbf{F} = [\mathbf{v}]_x$ from the FOE if the relative motion of cameras contains only translation (Hartley and Zisserman, 2003). If the fundamental matrix for two images is known, the closest valid affine transformation can be determined (Barath et al., 2016). These closest ones are labelled as ground-truth transformations in our experiments. The details of the comparison using the fundamental matrix can be found in Sec. 4.2.

3.3 Uniform Motion: Bottom View

This motion is the same as described in the previous section. However, the camera observes the ground, instead of facing forward. Since the movement is parallel to the image plane, it can be considered as a degenerative case of the previous one, because the FOE is located at the infinity. In this scenario the features of the ground are related by a pure translation between the images as the ground is planar. The projections of the same corresponding features form a line in the camera images, but these epipolar lines are parallel to each other, and also parallel to the motion of the quadcopter. Fig. 4 shows example images of this motion and colored boxes related by the affine transformations.

Let us denote the angle between the epipolar lines and the X axis by γ , and l^i denotes a point which lies on the i -th epipolar line. The cost function to be minimized contains the squared distances of the measured points to the related epipolar lines:

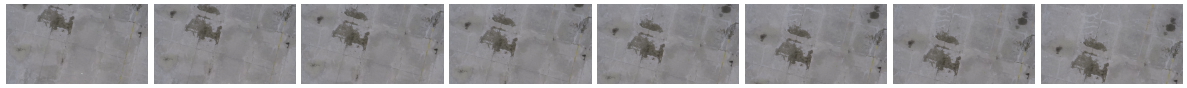
$$\sum_{f=1}^F \sum_{i=1}^P \left((\mathbf{p}_i^f - \mathbf{l}^i)^T \begin{bmatrix} -\sin \gamma \\ \cos \gamma \end{bmatrix} \right)^2. \quad (17)$$

The alternation minimizes the error with refining the angle of the epipolar lines (γ) first, then their points (\mathbf{l}^i). The first part of the alternation can be written as a homogeneous system of equations $\mathbf{A} \mathbf{x} = 0$ with respect to $\mathbf{x}^T \mathbf{x} = 1$, similarly to Eq. 13, but it contains all points for all images:

$$\begin{bmatrix} x_1^f - l_x^1 & y_1^f - l_y^1 \\ \vdots & \vdots \\ x_2^f - l_x^2 & y_2^f - l_y^2 \end{bmatrix} \begin{bmatrix} -\sin \gamma \\ \cos \gamma \end{bmatrix} = 0, \quad (18)$$

the solution is obtained as the eigenvector of the smallest eigenvalue of matrix $\mathbf{A}^T \mathbf{A}$, then $\gamma = \text{atan2}(v_x, v_y)$.

The second step of the alternation refines the points located on the epipolar lines. The equation can



(a) Example images for uniform motion, bottom view.



(b) The colored boxes indicates the same areas computed by the motion parameters.

Figure 4: The images are taken while the quadcopter is moving forward.

be formed as $Ax = b$, where

$$\begin{bmatrix} \cos \gamma & \sin \gamma \\ \vdots & \vdots \\ \cos \gamma & \sin \gamma \end{bmatrix} \begin{bmatrix} j_x^i \\ j_y^i \end{bmatrix} = \begin{bmatrix} x_i \cos \gamma + y_i \sin \gamma \\ \vdots \\ x_i \cos \gamma + y_i \sin \gamma \end{bmatrix}, \quad (19)$$

the point on the line is given by the pseudoinverse of

$$\mathbf{A}, \mathbf{I} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A} \mathbf{b}.$$

Ground Truth Affine Transformations The GT affine transformations for forward motion with a bottom-view camera is a simple identity:

$$\mathbf{A} = \mathbf{I} \quad (20)$$

3.4 Scaling

This motion is generated by the sinking of the quadcopter, while the camera observes the ground. The direction of the motion is approximately perpendicular to the camera plane, thus this can be considered as a special case of the Uniform Motion: Front View. The only difference is that the ground can be considered as a plane, thus the parameters of the motion and the affine transformation can be precisely calculated. Fig. 5 first row shows an example images captured during the motion.

Because of the direction of the movement and the camera plane are not perpendicular, the FOE and the epipolar lines for the corresponding selected features can be computed. It is also true, that during the sinking of the quadcopter, the features move along their corresponding epipolar line. See Seq. 3.2 for the computation of the FOE and epipolar lines.

The corresponding features are related in the images by the parameter of the scaling. Let s be the scaling parameter, then the distances between the features and the FOE are related by s . This can be formalized as follows:

$$(\mathbf{p}_i^2 - \mathbf{v}) = s_2 (\mathbf{p}_i^1 - \mathbf{v}) \quad i \in [1, P]. \quad (21)$$

Thus, the parameter of the scale is given, by the average of the fraction of the distances:

$$s_2 = \frac{1}{P} \sum_{i=1}^P \frac{\|\mathbf{p}_i^2 - \mathbf{v}\|_2}{\|\mathbf{p}_i^1 - \mathbf{v}\|_2}. \quad (22)$$

Ground Truth Affine Transformations. The GT affine transformations for the scaling is trivially a simple scaled identity:

$$\mathbf{A}_j = s \mathbf{I} \quad (23)$$

4 EVALUATION METHOD

The evaluation of the feature detectors is twofold. In the first comparison, the detection of features location and affine parameters are evaluated. To compare the feature locations and affine parameters, the camera motion needs to be known. These can be calculated for each motion described in the previous section, however, for the Uniform Motion: Front View, it is not possible. Thus, a second comparison is carried out, which uses only the fundamental matrix instead of the motion parameters.

4.1 Affine Evaluation

Location error. The location of the GT feature can be determined by the location of the same feature in the previous image, and the parameters of the motion. The calculation of the motion parameter differs from motion to motion, the details can be found in the related sections. The error of the feature detection is the Euclidean distance of the GT and the estimated feature:

$$Err_{det}(\mathbf{P}_{estimated}, \mathbf{P}_{GT}) = \|\mathbf{P}_{estimated} - \mathbf{P}_{GT}\|_2^2, \quad (24)$$

where $\mathbf{P}_{GT}, \mathbf{P}_{estimated}$ are the GT and estimated feature points, respectively.

Affine Error. While the error of the feature detection is based on the Euclidean distance, the error of affine transformation is calculated using the Frobenius norm of the difference matrix of the estimated and GT affine transformation. It can be formalized as follows:

$$Err_{aff}(\mathbf{A}_{estimated}, \mathbf{A}_{GT}) = \|\mathbf{A}_{estimated} - \mathbf{A}_{GT}\|_F, \quad (25)$$

where $\mathbf{A}_{GT}, \mathbf{A}_{estimated}$ are the GT and estimated affine transformations, respectively. The Frobenius norm



(a) Example images for the scaling test.



(b) The colored boxes indicate the same areas computed by the scaling parameters.

Figure 5: The images are taken while the quadcopter is sinking.

is chosen, because it has a geometrical meaning of the related affine transformations, see (Barath et al., 2016) for details.

4.2 Fundamental Matrix Evaluation

For the motion named as Uniform Motion: Front View, the affine transformations and parameters of the motion can not be estimated. The objects, which are closer to the camera move more pixels between the successive images, than objects located at the distance. Only the direction of the moving features can be calculated. That is the epipolar line that goes through the feature and the FOE. However, the fundamental matrix can be precisely calculated from FOE, $\mathbf{F} = [\mathbf{v}]_x$ and it can be used to refine the affine transformations.

The refined affine transformations are considered as GT, and used for the calculation of affine errors, as it is described in the previous section. (Barath et al., 2016) introduced the algorithm to find the closest affine transformations corresponding to the fundamental matrix. The paper states that it can be determined by solving a six-dimensional linear problem if the Lagrange-multiplier technique is applied for the constraints for the fundamental matrix.

For quantifying the quality of an affine transformation, the closest valid affine transformation is computed first by the method of (Barath et al., 2016). Then the difference matrix between the original and closest valid affine transformation is computed. The quality is given by the Frobenius norm of this difference matrix.

5 COMPARISON

Eleven affine feature detectors have been compared in our tests. The implementations are downloaded from the website of Visual Geometry Group, University of Oxford², except the TBMR, which is available on the

²www.robots.ox.ac.uk/vgg/research/affine/index.html

website of the author³. Most of the methods are introduced in Sec. 2. Additionally to those, HARHES and SEDGELAP are added to the comparison. HARHES is the composition of HARAFF and HESAFF, while SEDGELAP finds shapes along the edges, using the Laplace operator.

After the features are extracted from the images, feature matching is done considering SIFT descriptors. The ratio test published in (Lowe, 2004) was used for outlier filtering. Finally, the affine transformations are calculated for the filtered matches. The detectors determine only the elliptical area of the affine shapes, without orientation. The orientation is assigned to the areas using the SIFT descriptor in a separate step. Finally, the affine transformation of the matched feature is given by:

$$\mathbf{A} = \mathbf{A}_2 \mathbf{R}_2 (\mathbf{A}_1 \mathbf{R}_1)^{-1}, \quad (26)$$

where $\mathbf{A}_i, \mathbf{R}_i, i \in [1, 2]$ are the local affine areas defined by ellipses, and the related rotation matrices assigned by the SIFT descriptor, respectively.

Table 1 summarizes the number of features, number of matched features and running time of the detectors on the tests. SEDGELAP and HARHES find the most features, however, the high number of features makes the matching more complicated and time consuming. In general, EBR, IBR and MSER find hundreds of features, the Harris based methods (HARAFF, HARLAP) find approximately ten to twenty thousand, and the Hessian based methods (HESAFF, HESLAP) find a few thousands of features. The ratio test (Lowe, 2004), used for outlier filtering, excludes some feature matches. The second row of each test sequence in Table 1 shows the number of features after matching and outlier filtering. Note, that approximately 50% of features are lost due to the ratio test. The running time of the methods are shown in the third rows of each test sequence. These times highly depend on the image resolution, which is 5MP in our tests. Each implementation was run on the CPU, using one core of the machine. Obviously, MSER is the fastest method, followed by SURF and

³<http://laurentnajman.org/index.php?page=tbmr>

Table 1: Number of features, number of matched features and required running time for test sequences. The columns are the methods and the rows (triplets) are the test sequences. **First row** of each sequence shows the number of detected features. The number of matched features are shown in the **second row**. The **third row** of each test sequence contains the required running times.

		EBR	HARAFF	HARHES	HARLAP	HESAFF	HESLAP	IBR	MSER	SEDGELAP	SURF	TBMR
Scaling	# All	217	18720	21198	19033	3866	3994	1277	336	20132	783	4122
	# Matched	91	8369	9544	8534	1702	1757	678	235	10071	636	2259
	Running time (s)	20.43	3.27	2.32	2.06	1.19	0.97	7.59	0.41	3.29	0.65	0.68
Rotation	# All	114	12686	14266	12795	2655	2751	892	223	13173	508	3141
	# Matched	33	5235	5943	5294	1070	1111	395	133	5872	390	1447
	Running time (s)	18.01	2.21	1.69	1.56	0.91	0.77	6.27	0.35	2.41	0.54	0.54
Bottomview1	# All	59	7514	8122	7629	1211	1216	441	92	7222	204	2688
	# Matched	27	3450	3730	3503	522	524	226	64	3743	169	1049
	Running time (s)	18.09	1.69	1.37	1.30	0.80	0.73	5.52	0.38	1.85	0.51	0.67
Bottomview2	# All	26	12484	13020	12615	1119	1135	588	117	9461	139	3051
	# Matched	8	4451	4681	4506	418	426	241	68	4094	107	1037
	Running time (s)	20.49	2.22	1.65	1.64	0.78	0.72	5.25	0.36	2.11	0.57	0.95
FrontView1	# All	78	10128	13639	10415	4708	5032	695	233	14869	803	2083
	# Matched	36	4623	6476	4758	2444	2583	416	164	7650	653	1058
	Running time (s)	17.97	3.16	2.12	1.66	1.70	1.19	8.47	0.29	2.67	0.81	0.76
FrontView2	# All	59	9210	11193	9369	2108	2812	592	50	12345	511	1856
	# Matched	30	4615	5764	4703	1192	1617	398	38	7275	447	1034
	Running time (s)	15.56	2.07	1.65	1.48	1.01	0.87	6.58	0.29	2.31	0.65	0.55

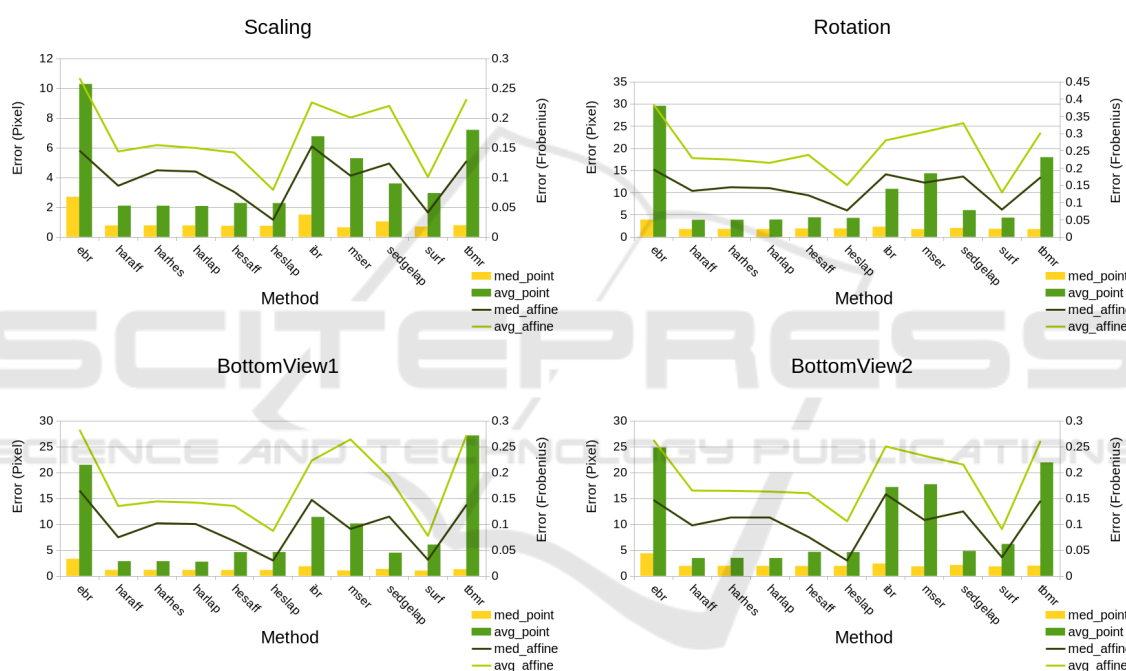


Figure 6: The error of affine evaluation. The error of feature detection is measured in pixels, and can be seen on the left axis. The error of affine transformations is given by the Frobenius norm, it is plotted on the right axis. The average and median values for both the affine and detection errors are shown.

TBMR. The Hessian based methods need approximately 1 second to process an image, while Harris based methods need 1.5 or 2 more times. The slowest are IBR and EBR. Note that, by parallelism and/or GPU implementations, the running times may show different results.

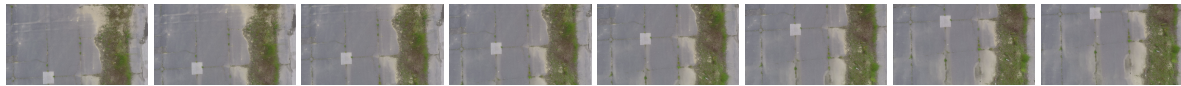
5.1 Affine Evaluation

The first evaluation uses the estimated camera motions and affine transformation introduced in Sec. 3. The quantitative evaluation is twofold, since the localization of features and accuracy of affine transfor-

mation can both be evaluated.

Four test sequences are captured. One for the scaling, one for the rotation and two for the Uniform Motion: Bottom View motions. See Fig. 2 for the rotation, Fig. 5 for the scale, first row of Fig. 4 and Fig. 7 for the Uniform Motion: Bottom View test images.

Fig. 6 summarizes the errors for the detectors. The average and median error of feature detection can be seen on the bar-charts, where the left vertical axis mark the Euclidean distance between the estimated and GT feature, measured in pixels. The average and median error of the affine transformations are visualized as green and black lines, respectively. The mea-



(a) Example images for the BottomView2 test.



(b) Example images for the FrontView test.

Figure 7: The images are taken while the quadcopter is moving parallel to the ground. **First Row:** The camera observes the ground. **Second row:** It faces to the front.

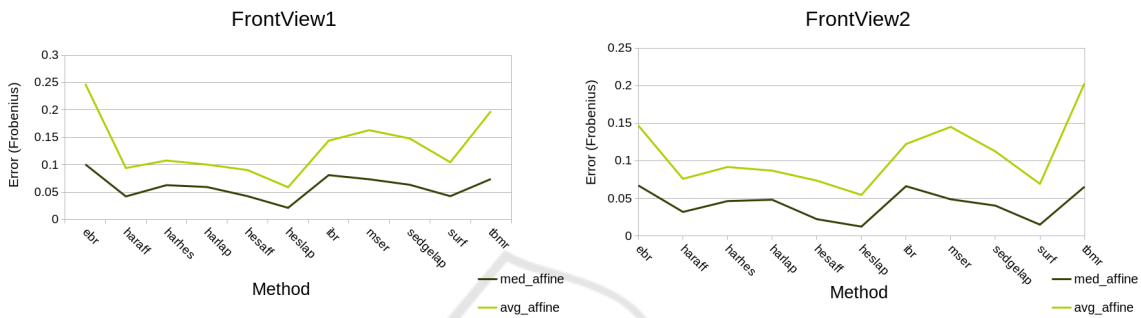


Figure 8: The error of Fundamental Matrix Evaluation.

sure is given by the Frobenius norm of the difference of the GT and estimated affine transformations. The quantity of the error can be seen at the right vertical axis of the charts.

The charts of Fig. 6 indicate similar results for the test sequences. The detection error is always the lowest for the Hessian and Harris based methods in average, indicating that these methods can find features more accurately than others. This measure is the highest for the EBR, IBR, MSER and TBMR. The averages can even be higher than 10 pixels, except for the scaling test case. Remark that the median values are always lower than the averages, because despite of the outlier filtering, the false matches can yield large detection error, which distorts the averages. Surprisingly, the affine error of HESLAP and SURF is the lowest, while that of the EBR, IBR and TBMR is the highest in all test cases.

5.2 Fundamental Matrix Evaluation

In case of the fundamental matrix evaluation, the quadcopter moves forward, perpendicular to the image plane. Objects are located at different distances from the camera, thus the GT affine transformation and GT position of features can not be recovered. In this comparison, the fundamental matrix is used to refine the estimated affine transformations. Then, the error is measured by Frobenius norm of the difference matrix of the estimated and refined affine transforma-

tion.

Two test scenarios are considered. The example images of the first one can be seen in Fig. 3, where the height of the quadcopter was approximately 2 meters. The images of the second scenario are shown in the second row of Fig. 7, these images are taken at around the top of the trees.

Fig. 8 shows the error of the affine transformations. Two test sequences are captured, each shows similar result. The HARAFF, HARLAP and SURF affine transformation yield the least affine errors. The characteristic of the errors is similar to that in the previous comparison.

6 CONCLUSIONS

We have compared the most popular affine matcher algorithms in this paper. The main novelty of our study is that the comparisons have been carried out on realistic images taken by a quadcopter. Our test sequences consists of more complex test cases than a simple homography estimation: rotation and scaling appear in the test as well. As a side effect, point matchers has also been compared as affine matching is impossible without point matching.

The most important conclusion of the tests that the performance of the affine detectors do not depend on the type of the sequence. Based on the results, the authors of this paper suggest to apply Harris-based,

Hessian-based and SURF algorithm to retrieve high quality affine transformations from image pairs.

ACKNOWLEDGEMENTS

EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies – The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

REFERENCES

- Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). Kaze features. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 214–227, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Barath, D., Matas, J., and Hajder, L. (2016). Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*.
- Barath, D., Molnár, J., and Hajder, L. (2015). Optimal surface normal from affine transformation. In *VISAPP 2015 - Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Volume 3, Berlin, Germany, 11-14 March, 2015.*, pages 305–316.
- Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.
- Cordes, K., Rosenhahn, B., and Ostermann, J. (2013). High-resolution feature evaluation benchmark. In Wilson, R., Hancock, E., Bors, A., and Smith, W., editors, *Computer Analysis of Images and Patterns*, pages 327–334, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151.
- Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition.
- Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 2548–2555, Washington, DC, USA. IEEE Computer Society.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.
- Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In Heyden, A., Sparr, G., Nielsen, M., and Johansen, P., editors, *Computer Vision – ECCV 2002*, pages 128–142, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72.
- Pusztai, Z. and Hajder, L. (2017). Quantitative comparison of affine invariant feature matching. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) - Volume 6: VISAPP, Porto, Portugal, February 27 - March 1, 2017.*, pages 515–522.
- Shi, J. and Tomasi, C. (1994). Good features to track.
- Tareen, S. A. K. and Saleem, Z. (2018). A comparative analysis of sift, surf, kaze, akaze, orb, and brisk. In *International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*.
- Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280.
- Tuytelaars, T. and Van Gool, L. (2004). Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision*, 59(1):61–85.
- Xu, Y., Monasse, P., Graud, T., and Najman, L. (2014). Tree-based morse regions: A topological approach to local feature detection. *IEEE Transactions on Image Processing*, 23(12):5612–5625.
- Zitnick, L. and Ramnath, K. (2011). Edge foci interest points. International Conference on Computer Vision.

APPENDIX

The goal is to show how the following equation:

$$\mathbf{Ax} = \mathbf{b}$$

can be solved subject to $\mathbf{x}^T \mathbf{x} = 1$. The cost function must be written with the so-called Lagrangian multiplier λ . It is as follows:

$$J = (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) + \lambda \mathbf{x}^T \mathbf{x}.$$

The optimal solution is given by the derivative of the cost function w.r.t \mathbf{x} .

$$\frac{\partial J}{\partial \mathbf{x}} = 2\mathbf{A}^T (\mathbf{Ax} - \mathbf{b}) + 2\lambda \mathbf{x} = 0.$$

Therefore the optimal solution is as follows:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \mathbf{b}.$$

For the sake of simplicity, we introduce the vector $\mathbf{v} = \mathbf{A}^T \mathbf{b}$ and the symmetric matrix $\mathbf{C} = \mathbf{A}^T \mathbf{A}$, then:

$$\mathbf{x} = (\mathbf{C} + \lambda \mathbf{I})^{-1} \mathbf{v}.$$

Finally, the constraint $\mathbf{x}^T \mathbf{x} = 1$ has to be considered:

$$\mathbf{v}^T (\mathbf{C} + \lambda \mathbf{I})^{-T} (\mathbf{C} + \lambda \mathbf{I})^{-1} \mathbf{v} = 1.$$

By definition, it can be written that:

$$(\mathbf{C} + \lambda \mathbf{I})^{-1} = \frac{\text{adj}(\mathbf{C} + \lambda \mathbf{I})}{\det(\mathbf{C} + \lambda \mathbf{I})}.$$

If

$$\mathbf{C} = \begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix},$$

then

$$\mathbf{C} + \lambda \mathbf{I} = \begin{bmatrix} c_1 + \lambda & c_2 \\ c_3 & c_4 + \lambda \end{bmatrix}$$

The determinant and adjoint matrix of $\mathbf{C} + \lambda \mathbf{I}$ can be written as:

$$\det(\mathbf{C} + \lambda \mathbf{I}) = (c_1 + \lambda)(c_4 + \lambda) - c_2 c_3$$

and

$$\begin{aligned} \text{adj}(\mathbf{C} + \lambda \mathbf{I}) &= \begin{bmatrix} c_4 + \lambda & -c_2 \\ -c_3 & c_1 + \lambda \end{bmatrix} \\ \begin{bmatrix} c_4 + \lambda & -c_2 \\ -c_3 & c_1 + \lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} &= \begin{bmatrix} v_1 \lambda + c_4 v_1 - c_2 v_2 \\ v_2 \lambda + c_1 v_2 - c_3 v_1 \end{bmatrix}. \end{aligned}$$

Furthermore, the expression $\mathbf{v}^T (\mathbf{C} + \lambda \mathbf{I})^{-T} (\mathbf{C} + \lambda \mathbf{I})^{-1} \mathbf{v} = 1$ can be rewritten as

$$\mathbf{v}^T \frac{\text{adj}^T(\mathbf{C} + \lambda \mathbf{I}) \text{adj}(\mathbf{C} + \lambda \mathbf{I})}{\det(\mathbf{C} + \lambda \mathbf{I}) \det(\mathbf{C} + \lambda \mathbf{I})} \mathbf{v} = 1,$$

$$\mathbf{v}^T \text{adj}^T(\mathbf{C} + \lambda \mathbf{I}) \text{adj}(\mathbf{C} + \lambda \mathbf{I}) \mathbf{v} = \det^2(\mathbf{C} + \lambda \mathbf{I}).$$

Both sides of the equation contain polynomials. The degrees of the left and right sides are $2n - 2$ and $2n$, respectively. If the expression in the sides are subtracted by each other, a polynomial of degree $2n$ is obtained. Note that, $n = 2$ in the discussed case, i.e. planar motion. The optimal solution is obtained as the real roots of this polynomial. The vector corresponding to the estimated λ_i , $i = 1, 2$, is calculated as $\mathbf{g}_i = (\mathbf{L} + \lambda_i \mathbf{I})^{-1} \mathbf{r}$. Then the vector with minimal norm $\|\mathbf{F} \mathbf{g}_i - \mathbf{h}\|$ is selected as the optimal solution of the problem.