Accurate Reconstruction of Oriented 3D Points Using Affine Correspondences

Carolina Raposo[1,2(✉)] and Joao P. Barreto[1]

[1] Institute of Systems and Robotics, University of Coimbra, Coimbra, Portugal
{carolinaraposo,jpbar}@isr.uc.pt
[2] Perceive3D, Coimbra, Portugal

Abstract. Affine correspondences (ACs) have been an active topic of research, namely for the recovery of surface normals. However, current solutions still suffer from the fact that even state-of-the-art affine feature detectors are inaccurate, and ACs are often contaminated by large levels of noise, yielding poor surface normals. This article provides new formulations for achieving epipolar geometry-consistent ACs, that, besides leading to linear solvers that are up to 30× faster than the state-of-the-art alternatives, allow for a fast refinement scheme that significantly improves the quality of the noisy ACs. In addition, a tracker that automatically enforces the epipolar geometry is proposed, with experiments showing that it significantly outperforms competing methods in situations of low texture. This opens the way to application domains where the scenes are typically low textured, such as during arthroscopic procedures.

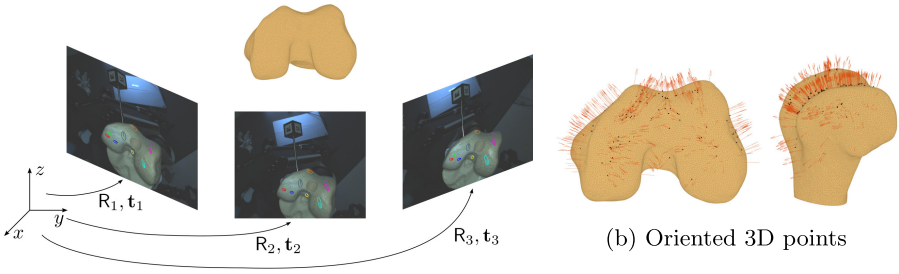
Keywords: Affine correspondences · Photoconsistency optimization · Tracking · Surface normal estimation

1 Introduction

Affine correspondences (ACs) encode important information about the scene geometry and researchers have been actively exploiting them for solving very different Computer Vision tasks, ranging from plane segmentation to the estimation of radial distortion parameters. In particular, Perdoch *et al.* [16] generate point correspondences from ACs for estimating the epipolar geometry, Bentolila and Francos [6] estimate the fundamental matrix from 3 ACs, Raposo and Barreto [18,20] use them to estimate the essential matrix and perform plane segmentation, Pritts *et al.* [17] retrieve distortion parameters from affine maps, and Hajder and Barath [9] accomplish planar motion estimation from a single AC. More recently, the estimation of affine transformations from two directions if the epipolar geometry is known has been proposed in [15].

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58604-1_33) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12373, pp. 545–560, 2020.
https://doi.org/10.1007/978-3-030-58604-1_33



(a) Affine multi-view tracks

(b) Oriented 3D points

Fig. 1. a) A calibrated camera whose pose is known at all times observes a 3D scene from different viewpoints. ACs extracted across multiple frames are shown and identified with colors. b) For each multi-view track of affine maps, the proposed method provides an oriented 3D point, i.e., its 3D location and normal to the surface. Reconstructed 3D points are shown in black and red arrows represent normals. (Color figure online)

The fact that an AC encodes information about the normal to the surface has motivated a series of recent works to estimate normals from ACs when the epipolar geometry is known [2–4,7,8], with important applications in the fields of object detection, 3D registration and segmentation. In general terms, two families of algorithms exist: one that estimates the surface normal directly from the extracted ACs [2,4,8] and another that starts by correcting the AC to be consistent with the epipolar geometry and afterwards retrieves the normal [3,7]. All solutions in the second family of algorithms perform an initial correction of the point correspondence and afterwards modify the affine transformation. When considering highly textured scenes, this two-step process is reliable since point correspondences are usually accurate and it is the affine component that is significantly affected by noise. However, when working in low textured scenes, it cannot be assumed that point correspondences are known or can be corrected by triangulation [10] since accurate ones are difficult to extract, with methods typically yielding very sparse and inaccurate reconstructions.

This article provides new insights on how to solve the problem of obtaining 3D oriented points, i.e. 3D points augmented with the information about the surface normal, from ACs. In addition, schemes for the refinement and tracking of ACs based on photoconsistency that automatically enforce the epipolar constraints and that work well in situations of low texture are proposed. A valid alternative would be to formulate the problem using the plane equation to represent homographies. However, working directly with ACs enables the extracted affine regions to be used as the integration domain in photoconsistency. In the case of homographies, the optimal integration region depends on the scene geometry, which is unknown and not straightforward to determine.

Being able to obtain accurate and rich reconstructions of 3D oriented points in low-texture scenes greatly benefits several monocular vision algorithms. Examples include the reconstruction of indoor scenes, which are typically dominated

by large low-textured planes, and the detection and 3D registration of objects with low texture. In particular, this would add significant value to the domain of arthroscopic procedures where obtaining 3D reconstructions of bone surface solely from the arthroscopic images is difficult mainly due to their inherent low texture [21]. In summary, the contributions are the following:

Fast Correction of ACs and Normal Estimation: Building on a recent study [20] that provides the relation between an AC and the epipolar geometry, we show how to write the AC as a function of only two unknown parameters (2 degrees of freedom (DoF)), in case the point correspondence is fixed, and three unknown parameters (3 DoF) otherwise, and propose fast linear solvers for enforcing noisy ACs to be consistent with the epipolar geometry. For the 2-DoF case, the multi-view solution comes in a straightforward manner. In addition, a fast linear solver for the estimation of normals from multiple frames is also presented.

Multi-view Refinement of ACs Consistent with the Epipolar Geometry: A fast method for multi-view photoconsistency refinement of the affine transformation of the AC that is the first to automatically enforce the epipolar geometry is proposed. Experiments show that it significantly improves the quality of the estimated normals, providing accurate oriented 3D points (Fig. 1).

Tracking of ACs Consistent with the Epipolar Geometry: We present the first formulation for correction of ACs to be consistent with the epipolar geometry that also corrects the point depth, avoiding the common two-step process [3] of fixing the point correspondence and the affine frame sequentially. Building on this formulation, a novel tracker that enables the accurate reconstruction of oriented 3D points in low textured scenes, outperforming a standard KLT tracker [1], is proposed.

2 Epipolar Geometry-Consistent ACs

Let $(\mathbf{x}, \mathbf{y}, \mathsf{A})$ be an affine correspondence (AC) such that the patches surrounding $\mathbf{x}$ and $\mathbf{y}$ are related by a non-singular 2×2 matrix A, with

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\mathsf{T}, \mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^\mathsf{T}, \mathsf{A} = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix}. \tag{1}$$

A point correspondence $(\mathbf{u}, \mathbf{v})$ in the patch is related by $\mathbf{v} = \mathsf{A}\mathbf{u} + \mathbf{b}$, with $\mathbf{b} = \mathbf{y} - \mathsf{A}\mathbf{x}$. As demonstrated in [20], an AC is consistent with the epipolar geometry if the following is verified:

$$\begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 & x_2 y_1 & x_2 y_2 & x_2 & y_1 & y_2 & 1 \\ a_3 x_1 & a_4 x_1 & 0 & y_1 + a_3 x_2 & y_2 + a_4 x_2 & 1 & a_3 & a_4 & 0 \\ y_1 + a_1 x_1 & y_2 + a_2 x_1 & 1 & a_1 x_2 & a_2 x_2 & 0 & a_1 & a_2 & 0 \end{bmatrix} \mathsf{E}(:) = \mathbf{0}, \tag{2}$$

with $\mathsf{D}(:)$ denoting the vectorization of matrix D by columns and E being the essential matrix.

From the relation $\mathbf{y} = \mathsf{A}\mathbf{x} + \mathbf{b}$, Eq. 2 can be written as

$$\mathsf{M}(\mathsf{E}, \mathbf{x}) \quad [\mathbf{m}^\mathsf{T} \quad 1]^\mathsf{T} = \mathbf{0}, \tag{3}$$

where M only depends on the essential matrix E and the point in the first image $\mathbf{x}$, and $\mathbf{m} = [\mathsf{A}(:)^\mathsf{T} \quad \mathbf{b}^\mathsf{T}]^\mathsf{T}$. By taking the null space of M, whose dimension is 3×7, a basis for the full AC $\underline{\mathsf{A}} = [\mathsf{A} \quad \mathbf{b}]$ is obtained, and thus $\underline{\mathsf{A}}$ can be written as a linear combination of this null space:

$$\underline{\mathsf{A}}(:) = \mathsf{N}_{6\times4} \quad [\underline{\alpha}^\mathsf{T} \quad 1]^\mathsf{T}, \tag{4}$$

with $\mathsf{N}_{6\times4}$ being the matrix that is obtained by removing the last row of the null space matrix of $\mathsf{M}_{3\times7}$ and $\underline{\alpha}$ being the set of unknown parameters $\underline{\alpha}^\mathsf{T} = [\underline{\alpha}_1 \quad \underline{\alpha}_2 \quad \underline{\alpha}_3]$. It comes in a straightforward manner that an AC $\tilde{\underline{\mathsf{A}}}$ extracted from a pair of images can be corrected to an AC $\underline{\mathsf{A}}$ that is consistent with the epipolar geometry by finding the solution $\underline{\alpha}^*$ to the system $\mathsf{N}_{6\times4}[\underline{\alpha}^{*\mathsf{T}} \quad 1]^\mathsf{T} = \tilde{\underline{\mathsf{A}}}(:)$ in the least-squares sense and afterwards computing $\underline{\mathsf{A}} = \mathsf{N}_{6\times4}[\underline{\alpha}^{*\mathsf{T}} \quad 1]^\mathsf{T}$.

As mentioned in the introduction, when extracting ACs in real scenarios, the level of noise present in the affine component is significantly larger than the one that affects the point correspondence. Thus, it may be desirable to assume that the point correspondence $(\mathbf{x}, \mathbf{y})$ is known, and only the affine component A is to be corrected to be consistent with the epipolar geometry. In this case, the problem is simplified since the two bottom equations of the system of Eqs. 2 can be written as

$$\mathsf{P}(\mathsf{E}, \mathbf{x}, \mathbf{y}) \quad [\mathsf{A}(:)^\mathsf{T} \quad 1]^\mathsf{T} = \mathbf{0}, \tag{5}$$

where P is a 2×5 matrix that only depends on the essential matrix E and the point correspondence $(\mathbf{x}, \mathbf{y})$. By taking the null space of P, the following is obtained

$$\mathsf{A}(:) = \mathsf{Q}_{4\times3}[\alpha^\mathsf{T} \quad 1]^\mathsf{T}, \tag{6}$$

where Q is the matrix that is obtained by removing the last row of the null space of P, having the following format

$$\mathsf{Q} = \begin{bmatrix} q_1 & 0 & q_2 \\ 1 & 0 & 0 \\ 0 & q_1 & q_3 \\ 0 & 1 & 0 \end{bmatrix}. \tag{7}$$

In this case, the number of degrees of freedom (DoF) is 2, i.e., $\alpha = [\alpha_1 \quad \alpha_2]^\mathsf{T}$. The corrected AC is estimated similarly to the 3-DoF case.

2.1 Extension to the Multi-view Case

2-DoF Formulation. Consider a 3-view affine track consisting of two ACs $(\mathbf{x}, \mathbf{y}, \mathsf{A})$ and $(\mathbf{y}, \mathbf{z}, \mathsf{B})$ that relate patches in frames 1 and 2 and frames 2 and 3, respectively. By assuming that the point correspondences are fixed, it is possible to correct ACs A and B independently by performing as previously described.

However, a multi-view formulation for correcting one AC using information from more than two views simultaneously yields more accurate results [7].

This section proposes a new linear solver for accomplishing this task. Let $(\mathbf{x}, \mathbf{z}, \mathsf{C})$ be the AC that relates the patch in frame 1 surrounding $\mathbf{x}$ with the patch in frame 3 surrounding $\mathbf{z}$, so that $\mathsf{C} = \mathsf{BA}$. By representing each AC as in Eq. 6, i. e., $\mathsf{A}(:) = \mathsf{Q}_\mathsf{A}[\alpha^\mathsf{T} \quad 1]^\mathsf{T}$, $\mathsf{B}(:) = \mathsf{Q}_\mathsf{B}[\beta^\mathsf{T} \quad 1]^\mathsf{T}$ and $\mathsf{C}(:) = \mathsf{Q}_\mathsf{C}[\gamma^\mathsf{T} \quad 1]^\mathsf{T}$, it is possible to write the unknown parameters β and γ as a function of α so that the latter can be estimated using the information from all three views:

$$\begin{aligned} \beta_1 &= \quad \lambda_1\alpha_1 + \lambda_2\alpha_2 / \lambda_3\alpha_1 + \lambda_4\alpha_2 \\ \beta_2 &= \lambda_5\alpha_1 + \lambda_6\alpha_2 + \lambda_7 / \lambda_3\alpha_1 + \lambda_4\alpha_2 \\ \gamma_1 &= \quad \lambda_8\alpha_1 + \lambda_9 \\ \gamma_2 &= \quad \lambda_8\alpha_2 + \lambda_{10} \end{aligned}, \tag{8}$$

where $\lambda_i, i = 1, \ldots, 10$ are parameters that only depend on the known matrices Q_A, Q_B and Q_C.

Since the relationship between γ and α is linear, a linear system of equations relating ACs A and C with the unknown parameters α,

$$\mathsf{L}[\alpha^\mathsf{T} \quad 1]^\mathsf{T} = [\mathsf{A}(:)^\mathsf{T} \quad \mathsf{C}(:)^\mathsf{T}]^\mathsf{T}, \tag{9}$$

can be written, where

$$\mathsf{L} = \begin{bmatrix} \mathsf{Q}_\mathsf{A} & \\ \lambda_8\mathsf{Q}_\mathsf{C}^{[1,2]} & \mathsf{Q}_\mathsf{C}[\lambda_9 \quad \lambda_{10} \quad 1]^\mathsf{T} \end{bmatrix}, \tag{10}$$

with $\mathsf{Q}_\mathsf{C}^{[1,2]}$ denoting columns 1 and 2 of matrix Q_C. This formulation can be extended to more than 3 views in a straightforward manner by performing similarly for each new frame and stacking the new equations to the linear system 9.

3-DoF Formulation. Performing multi-view correction of ACs in the general case, i.e., when it is not assumed that the point correspondences are known and thus the full AC $\underline{\mathsf{A}}$ is accounted for, is possible but not as simple as described for the 2-DoF case. The reason for this is that, when attempting to follow a procedure analogous to the 2-DoF case, since point $\mathbf{y}$ is not known, it becomes impossible to directly obtain a representation of $\underline{\mathsf{B}}$ as in Eq. 4. However, $\mathbf{y}$ can be written as $\mathbf{y} = \underline{\mathsf{A}}[\mathbf{x}^\mathsf{T} \quad 1]^\mathsf{T}$, which, together with the null-space representations of ACs $\underline{\mathsf{A}}$ and $\underline{\mathsf{C}}$,

$$\begin{aligned} \underline{\mathsf{A}}(:) &= \mathsf{N}_{\underline{\mathsf{A}}} \quad [\underline{\alpha}^\mathsf{T} \quad 1]^\mathsf{T} \\ \underline{\mathsf{C}}(:) &= \mathsf{N}_{\underline{\mathsf{C}}} \quad [\underline{\gamma}^\mathsf{T} \quad 1]^\mathsf{T}, \end{aligned} \tag{11}$$

yields, after considerable algebraic manipulation[1], the following system of equations

$$\mathsf{G}\,[\underline{\beta}^\mathsf{T} \; \underline{\gamma}^\mathsf{T}]^\mathsf{T} = \mathbf{g}, \tag{12}$$

[1] We used MATLAB's symbolic toolbox for performing the algebraic manipulation. The MATLAB code for deriving all the equations in this section is provided as supplementary material.

where G and $\mathbf{g}$ depend on $\underline{\alpha}$. Unfortunately, this dependency precludes a linear system such as the one in Eq. 9 from being obtained, making this formulation significantly more complex than the 2-DoF one. One possibility for achieving AC correction in this case is to devise an iterative scheme for minimizing the Frobenius norm of the difference between the extracted and the corrected ACs. This could be done by starting with an initialization for $\underline{\alpha}$ obtained from the extracted AC $\tilde{\underline{\mathsf{A}}}$, estimating $\underline{\beta}$ and $\underline{\gamma}$ using Eq. 12, retrieving the corrected ACs $\underline{\mathsf{A}}$, $\underline{\mathsf{B}}$ and $\underline{\mathsf{C}}$ and iterating for every new estimation of $\underline{\alpha}$ until the sum of the squared Frobenius norms of the differences between the extracted and the corrected ACs is minimal. Generalization to an arbitrary number of views comes in a straightforward manner.

3 Multi-view Linear Estimation of Surface Normals

It is well known that the affine transformation A of an AC is the Jacobian of the homography in point $\mathbf{x}$ [11,12,20]. This result enables to relate the AC with the normal to the surface at the corresponding 3D point $\mathbf{X}$, also enabling the latter to be estimated. Solutions for this problem, in the 2-view and multi-view cases, that formulate the problem in the 3D space have been proposed [2,4]. In this section we derive a simpler formulation that allows to build a linear solver for normal estimation in the multi-view case.

It has been shown in [20] that an AC $(\mathbf{x}, \mathbf{y}, \mathsf{A})$ induces a two-parameter family of homographies H that can be written up to scale as

$$\mathsf{H}(\mathbf{j}; \mathbf{x}, \mathbf{y}, \mathsf{A}) = \begin{bmatrix} \mathsf{A} + \mathbf{y}\mathbf{j}^\mathsf{T} & \mathbf{y} - (\mathsf{A} + \mathbf{y}\mathbf{j}^\mathsf{T})\mathbf{x} \\ \mathbf{j}^\mathsf{T} & 1 - \mathbf{j}^\mathsf{T}\mathbf{x} \end{bmatrix}. \tag{13}$$

The equality $\mathsf{H}(\mathbf{j}; \mathbf{x}, \mathbf{y}, \mathsf{A}) = \mathsf{R} + \mathbf{t}\mathbf{n}^\mathsf{T}$, where $\mathsf{R}, \mathbf{t}$ is the known rotation and translation between the cameras and $\mathbf{n}$ is the normal to be estimated, can be rewritten as $\mathsf{F}[\mathbf{n}^\mathsf{T} \quad \mathbf{j}^\mathsf{T}]^\mathsf{T} = -\mathsf{R}(:)$, with F being a 9×6 matrix that depends on $\mathbf{t}, \mathbf{x}, \mathbf{y}$ and A. By stacking the equations obtained for each view and solving the linear system, the multi-view estimation of $\mathbf{n}$ is accomplished.

Unlike in [2] where only the direction of the normal is recovered, this solver also provides the distance of the plane tangent to the surface, encoded in the norm of the normal vector. This extra information allows to reconstruct the 3D point by intersecting the back-projection rays of each camera with the plane.

4 Photoconsistency Optimization for Accurate Normals

Although there has been intensive research on affine region detectors [14,24], state-of-the-art methods still provide ACs that present high levels of noise [7]. Thus, in order to obtain accurate 3D oriented points, it does not suffice to correct the ACs to be consistent with the epipolar geometry. In this section, we propose two novel methods for photoconsistency error minimization that are based on the 2-DoF and 3-DoF formulations derived in Sect. 2. The first method works as

an optimizer of ACs and is designed to work in scenes with texture, where point correspondences can be accurately extracted. The second method is a tracker as it only requires feature detection in one frame and performs tracking for every incoming frame. It handles situations of low texture by performing the tracking constrained by the epipolar geometry.

4.1 2-DoF Formulation

The refinement of the affine component of the ACs is formulated as a non-linear optimization problem whose cost function is the photoconsistency error, i.e., the sum of the squared error between a template T, considered as the patch from the first frame that encloses the affine region, and the second frame I. Given an initial estimate for the parameters to optimize $\mathbf{p}$, the goal is to iteratively compute the update parameters $\delta\mathbf{p}$ by minimizing the cost function [1]

$$\sum_{x\in\mathcal{N}} \left[\underbrace{\mathrm{I}(w(\mathbf{x};\mathbf{p}+\delta\mathbf{p})) - \mathrm{T}(x)}_{E(\delta\mathbf{p})} \right]^2, \tag{14}$$

where w is the image warping function and $\mathcal{N}$ denotes the integration region.

The Efficient Second-order Minimization (ESM) alignment formulation [5,13] states that the incremental update $\delta\mathbf{p}$ which minimizes the error at each iteration is given, considering a second-order Taylor expansion approximation, by

$$\delta\mathbf{p} \approx -\left(\frac{J(\mathbf{0}) + J(\delta\mathbf{p})}{2}\right)^{+} E(\mathbf{0}), \tag{15}$$

where the symbol $^+$ denotes the pseudoinverse and $J(\mathbf{i})$ is the Jacobian of the error $E(\mathbf{i})$, having as general formula

$$J(\mathbf{i}) = \frac{\partial E(\mathbf{i})}{\partial \mathbf{i}} = \frac{\partial \mathrm{I}\left(w\left(\mathbf{x};\mathbf{p}+\mathbf{i}\right)\right)}{\partial \mathbf{i}}. \tag{16}$$

The Jacobian $J(\mathbf{0})$ evaluated using the current solution is given by

$$J(\mathbf{0}) = \left.\frac{\partial \mathrm{I}\left(w\left(\mathbf{x};\mathbf{p}+\mathbf{i}\right)\right)}{\partial \mathbf{i}}\right|_{\mathbf{i}=\mathbf{0}} = \left.\frac{\partial \mathrm{I}(\mathbf{x}')}{\partial \mathbf{x}'}\right|_{\mathbf{x}'=w(\mathbf{x};\mathbf{p})} \left.\frac{\partial w\left(\mathbf{x};\mathbf{p}+\mathbf{i}\right)}{\partial \mathbf{i}}\right|_{\mathbf{i}=\mathbf{0}}. \tag{17}$$

The first term on the right-hand side of Eq. 17 is the gradient of the image warped at the current solution. The second term is the Jacobian of the warp function evaluated at $\mathbf{i} = \mathbf{0}$ which, using the formulations derived in Sect. 2, is easy to compute. For the sake of computational efficiency, we obtain the incremental update by solely considering $J(\mathbf{0})$, i.e., by computing $\delta\mathbf{p} = -J(\mathbf{0})^{+}E(\mathbf{0})$, which is a valid approximation.

In the present case, where the point correspondence $(\mathbf{x}, \mathbf{y})$ is fixed, the unknown parameters $\mathbf{p}$ to be refined correspond to α in Eq. 6 and the warp

function w transforms points $\mathbf{u}$ in the template into points $\mathbf{v}$ in the second image by an affine projection $\mathsf{H}_\mathsf{A} = [\mathsf{A}(\mathsf{Q}_\mathsf{A}, \mathbf{p}) \quad \mathbf{y} - \mathsf{A}(\mathsf{Q}_\mathsf{A}, \mathbf{p})\mathbf{x}]$, where $\mathsf{A}(\mathsf{Q}_\mathsf{A}, \mathbf{p})$ is the 2×2 matrix computed using Q_A and $\mathbf{p}$, as described in Sect. 2.

The extension to the multi-view case is obtained by writing the warp function that transforms points $\mathbf{u}$ in the template into points $\mathbf{w}$ in the third image as a function of the unknown parameters $\mathbf{p}$ using the relation between α and γ derived in Eq. 8. The Jacobian of this warp function is then computed, as well as the gradient of the third image warped at the current solution. By stacking the errors E and their Jacobians, obtained using frames 2 and 3, the update $\delta\mathbf{p}$ is computed using the information of the 3 frames simultaneously. By performing similarly for every incoming frame, the multi-view photoconsistency refinement of the affine transformation A is achieved.

4.2 3-DoF Formulation

The formulation for the case of 3 unknown parameters is analogous to the 2-DoF case, with the unknown parameters vector $\mathbf{p}$ corresponding to $\underline{\alpha}$ in Eq. 4, and the warp function being determined using matrix N. Since in this case the unknown parameters $\mathbf{p}$ allow to optimize both the affine component and the translation part, this formulation can be used as a tracker, with the affine features being extracted in the first frame for creating the templates to be tracked.

As previously explained, one drawback of this formulation is that, since it is not possible to obtain a linear relation between $\underline{\alpha}$ and $\underline{\gamma}$, as in the 2-DoF case, this formulation cannot be extended to the multi-view case in a straightforward manner. However, an alternative formulation for minimizing the cost function 14 can be devised using non-linear optimization algorithms such as Levenberg-Marquardt and the relation between $\underline{\alpha}$, $\underline{\beta}$ and $\underline{\gamma}$ derived in Eq. 12.

5 Experimental Validation

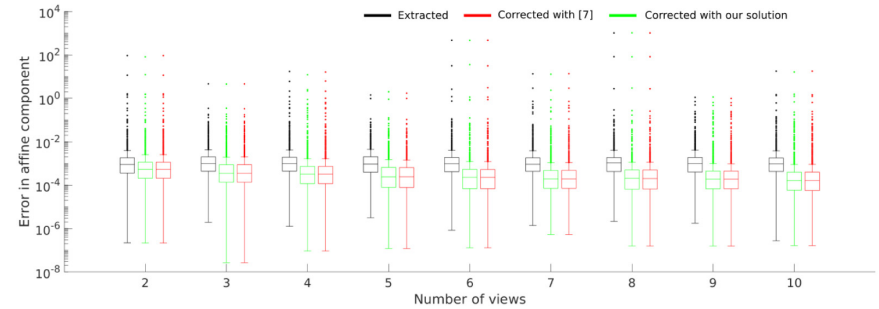
In this section, the proposed algorithms for AC correction, normal estimation, refinement of ACs using photoconsistency and tracking of affine regions are tested and compared with the state-of-the-art methods, both using synthetic data and real-world datasets. In all experiments using real data, affine covariant features are extracted with the Hessian Laplace detector [14,24] using the VLFeat library [25].

5.1 Synthetic Data

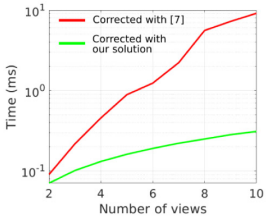
This experiment serves to compare the accuracy and computational efficiency of the two proposed linear solvers for correcting ACs and estimating normals with the state-of-the-art solutions [7] and [2], respectively.

The synthetic setup was generated as described in [2,7][2], consisting of N cameras randomly located on the surface of a sphere of radius 5 and looking

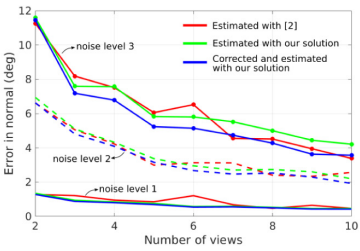
[2] We thank the authors for kindly providing the source code.



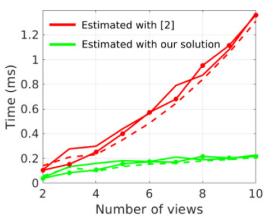
(a) Error in affine component



(b) Time (ms) for a C++ implementation



(c) Error in normal (deg)



(d) Time (ms) for a Matlab implementation

Fig. 2. Comparison of the proposed multi-view solver for correcting ACs (a and b) and the proposed multi-view method for normal estimation (c and d) with the state-of-the-art alternatives [7] and [2], respectively. a) The proposed and competing methods provide the same solutions when performing AC correction, with b) our method being over an order of magnitude faster for $N > 5$ views. c) Similar results are obtained by both methods, with [2] being slightly more accurate for $N > 6$ and the highest noise level. Correcting the AC prior to estimating the surface normal is systematically the best option. d) While our method scales well, having a nearly constant computational time for increasing number of views, [2] presents higher computational times that increase approximately linearly with the number of views.

towards the origin of that sphere. Random oriented 3D points were generated and projected onto the cameras, allowing the estimation of ground truth affine maps and point locations. Zero-mean Gaussian noise with standard deviation σ was added to the affine components. Figure 2 gives the comparison of our solvers with the ones presented in [2,7], in terms of accuracy and computational time. The ACs correction solvers are implemented in C++, while the normal estimation algorithms are implemented in Matlab. The number of views N varies from 2 to 10 and different noise levels are considered, by varying σ. Results were obtained over 1000 trials.

Figure 2a shows the distribution of errors in the affine components of the extracted ACs and of both the ACs corrected with the method proposed in [7] and our approach. The error in the affine component is computed as the Frobenius norm of the difference between each AC 2×2 matrix and the ground truth

(a) (b) (c) (d) (e)

Fig. 3. Datasets used in the photoconsistency refinement experiment. The datasets consist of images acquired in very different scenes and the camera poses come from distinct sources, including detection of fiducial markers (a and b), application of an SfM pipeline (c and d) and GPS measurements (e). The high variability of the datasets evinces the large range of possible applications of the proposed approach. The datasets are identified in Fig. 4 by a) Bone model, b) Bone model arthro, c) herzjesu-p8, d) fountain-p11 and e) freiburg3.

one. It can be seen that our proposed solver provides the exact same solution as [7], while being significantly faster, as shown in Fig. 2b that it achieves a speed up of over 30× for $N = 10$. While the solution in [7] involves computing SVD of a $2N \times C$-matrix, with C being the combination of all pairs of views, and performing two multiplications of matrices with considerable size, our solution solely requires the computation of the SVD of a $4(N-1) \times 3$ matrix. As an example, for $N = 10$, the matrices sizes are 20×45 ([7]) vs 36×3 (ours). This difference in the solver results in a dramatic decrease in computational times. In addition, it can be seen that for the considered noise level ($\sigma = 1$) correcting the ACs always makes them closer to the ground truth ones.

In order to compare the performance of the multi-view normal estimation algorithm presented in [2] with our linear solver, we fed both algorithms with the noisy ACs and computed the angle between the obtained normals and the ground truth ones. Results are shown in Fig. 2c, where the angular errors of the normals estimated after correcting the ACs are also plotted. In this case, since the two solvers provide different solutions, we tested for different noise levels by considering $\sigma = 0.2, 1, 2$. It can be seen that although the solutions are not identical, they are very similar, demonstrating the effectiveness of our proposed linear solver. This result also confirms the findings reported in [7] that correcting the ACs before estimating the surface normal is beneficial in almost every case. Regarding the computational time, our solver is about 6.5× faster than [2] for 10 views, and, unlike the latter, scales well for increasing number of views. The reason for this considerable speed up is that while the number of equations in our normal estimation solver is equal to $9(N-1)$, the complexity of the one presented in [2] increases quadratically with N.

5.2 Photoconsistency Refinement

In this experiment we evaluate our proposed algorithm for optimizing ACs based on photoconsistency by considering 5 datasets of very different scenes for which dense 3D models exist, and containing images for which the cameras' poses are known, as well as their intrinsic calibrations. For each dataset, multi-view tracks

Table 1. Average times in ms of Matlab implementations of the proposed **Ref2DoF** method and **Ref4DoF**, for different number of views.

# views		2	3	4	5
Avg. time (ms)	**Ref2DoF**	19.7	39.7	62.0	103.6
	Ref4DoF	36.3	71.9	108.8	168.5

of affine maps were extracted and the ground truth 3D points and normals were obtained by intersecting the back-projection rays of the first camera with the 3D model, and retrieving both the point of intersection and the normal at that point. In order to enforce the epipolar geometry assumptions, for each multi-view track, the point was triangulated [10] and projected onto each camera, yielding correspondences that perfectly satisfy the epipolar geometry.

The considered datasets, described in Fig. 3, are the two sequences from the Strecha dataset [22] *fountain-p11* and *herzjesu-p8* with publicly available ground truth 3D point cloud, the sequence *freiburg3 nostructure texture far* from the RGB-D SLAM Dataset and Benchmark [23] and two other sequences we acquired similarly to what is described in [21]. In more detail, we considered a 3D printed model of a bone to which a fiducial with printed binary square patterns is attached and can be tracked, providing the camera pose for every frame. We acquired two sequences, one with a large-focal distance lens and another with an arthroscopic lens. For both sequences we undistorted the images before extracting ACs.

The proposed approach, referred to as **Ref2DoF**, is compared with 4 alternative methods: (1) estimating the normals directly from the extracted ACs, (2) correcting the ACs and afterwards estimating the normals, (3) performing a photoconsistency refinement using a 4-DoF formulation (referred to as **Ref4DoF**), where all 4 parameters of the affine transformation are considered as unknown parameters, and applying (1), and (4) performing (3) followed by (2).

Figure 4 shows the angular errors of the normals obtained by all 5 methods for the different datasets, and Table 1 gives the runtimes for **Ref2DoF** and **Ref4DoF**, for a varying number of views. Results show that although **Ref4DoF** significantly improves the quality of the estimated normals, it is always less accurate than our 2-DoF refinement algorithm, while also being considerably slower. In addition, it can be seen that, as expected, the improvement obtained by correcting the ACs is irrelevant when compared to the one achieved by a photoconsistency refinement. This experiment shows not only that refining ACs is crucial for achieving accurate 3D oriented points, but also that incorporating the constraints of the epipolar geometry into the refinement benefits both the accuracy and the computational efficiency. Figure 1b depicts some of the 3D oriented points obtained on the bone model dataset, where it can be visually confirmed that the normals are nearly perpendicular to the surface.

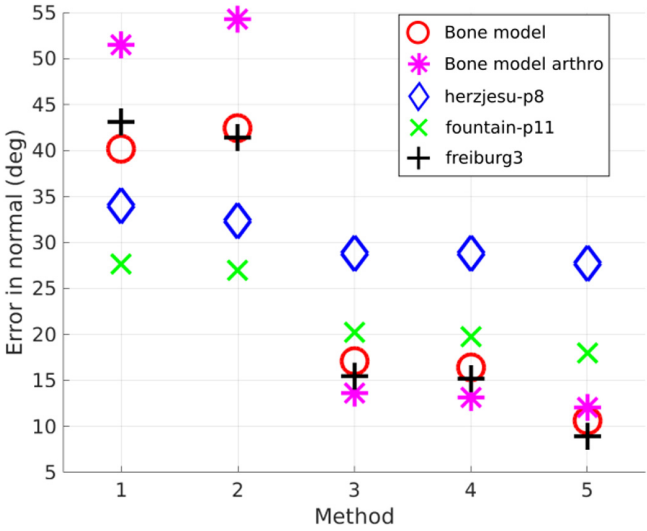
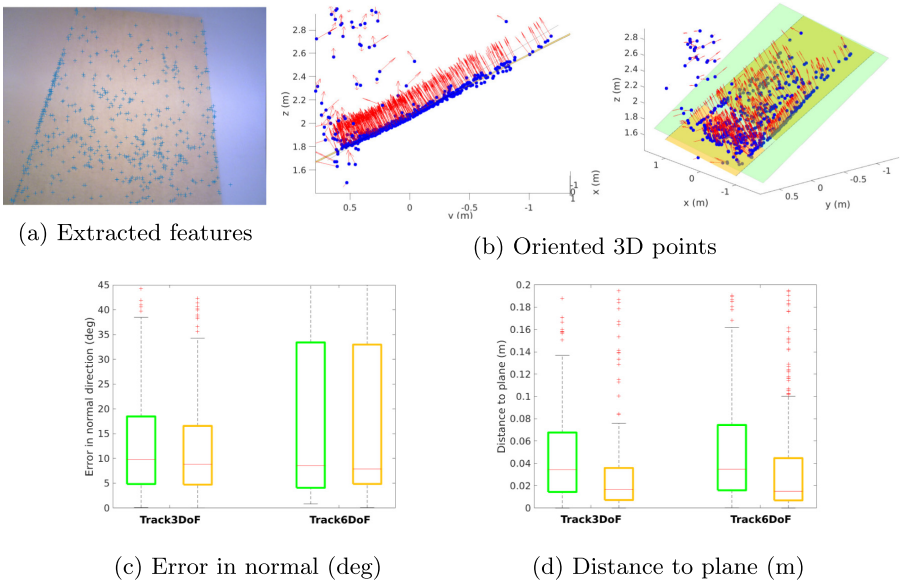


Fig. 4. Average errors in the normals estimated by 5 alternative methods, on the datasets described in Fig. 3. Methods 1 to 5 correspond to: (1) - estimating the normals directly from the extracted ACs; (2) - correcting the ACs and afterwards applying (1); (3) - **Ref4DoF** + (1); (4) - **Ref4DoF** + (2); (5) - **Ref2DoF**

5.3 Tracking

This final experiment serves to assess the performance of our proposed 3-DoF tracker (**Track3DoF**) under very challenging conditions of low texture, where existing solutions perform poorly. For this, we selected the sequence *freiburg3 nostructure notexture far* from the RGB-D SLAM Dataset and Benchmark [23] and extracted affine covariant features from the first frame. Figure 5a shows the frame with the point locations of the features. These features were tracked across 10 frames both using the proposed method **Track3DoF** and a formulation using 6 DoFs, referred to as **Track6DoF**, which is equivalent to a standard KLT affine tracker [1]. Figure 5b shows the obtained 3D oriented points by **Track3DoF**, as well as a green and a yellow planes. The green plane is obtained by finding the plane that best fits to the point cloud provided by the depth camera, and the yellow plane is obtained similarly from the reconstructed points. In order to quantitatively assess the quality of the oriented 3D points, we computed the angle between each obtained normal and the normals of both the green and the yellow planes, and the distance of each 3D point to both planes. Results are shown in Figs. 5c and 5d, and also include the errors obtained for **Track6DoF**, which were computed in a similar manner. It can be seen that 75% of the normals estimated by our method have an error below 20°, while for **Track6DoF** the value for the third quartile is 1.8× larger. Also, while our approach managed to successfully track 76% of the features, the 6-DoF formulation yielded only 64.8% of tracks with symmetric epipolar distance below 5 pix. In terms of computational time,



(a) Extracted features

(b) Oriented 3D points

(c) Error in normal (deg)

(d) Distance to plane (m)

Fig. 5. Experiment of tracking features in very low texture conditions. a) Affine features extracted from the first image to be tracked across multiple images. b) Oriented 3D points, represented as blue spheres with red arrows, reconstructed using the proposed **Track3DoF** method, with 76% of the features being correctly tracked. The obtained 3D points are fitted to a plane (yellow plane), as well as the ground truth 3D points provided in the dataset (green plane). Since the depth sensor presents non-negligible noise, we computed the errors in (c) the normals and in (d) the 3D point locations for both planes. Our method outperforms a standard 6-DoF affine KLT formulation (**Track6DoF**) that only successfully tracks 64.8% of the features and is 1.25× slower. (Color figure online)

our formulation is 1.25× faster, taking on average 45ms per tracklet in a Matlab implementation. In addition, we attempted to perform feature matching with the other frames in the dataset but, in this case, most retrieved correspondences were incorrect, and only 29% yielded a symmetric epipolar distance below 5 pix.

These experimental results confirm that including information about the epipolar geometry in the estimation of oriented 3D points significantly improves their quality. In particular, when working in very low textured situations, where feature matching algorithms fail and standard 6-DoF trackers perform poorly, our proposed solution is a viable alternative.

6 Conclusions and Future Work

We investigate the use of ACs in the tasks of normal estimation and reconstruction of oriented 3D points. Existing solutions still suffer from the low accuracy of affine detectors, yielding normals that are far from the ground truth. This

paper proposes methods that greatly improve the quality of noisy ACs, being an advance in the literature on this subject and also having practical relevance.

We provide new, simpler representations for ACs consistent with the epipolar geometry. As a consequence, we obtain a multi-view AC correction linear solver that outperforms the state-of-the-art in terms of computational time for any number of views, reaching a speed up of 30× for the case of 10 views. A novel linear solver for the multi-view estimation of normals is also presented, and experiments demonstrate that it is a valid faster alternative to the existing solvers. The novel simple representation of epipolar geometry-consistent ACs enables refinement schemes to be formulated as photoconsistency-based trackers, which, as demonstrated by the experimental results, significantly improve the quality of the extracted ACs. In addition, another important contribution of this paper is the new 3-DoF tracker that works in scenes presenting low texture, which is faster and accurately tracks more features than the standard affine 6-DoF KLT tracker.

The proposed 3-DoF tracker opens the way to applications in new domains. One important area where this type of tracker would be very useful is in surgical arthroscopic procedures, such as the reconstruction of the anterior cruciate ligament in the knee joint or the resection of the femoroacetabular impingement in the hip joint, where the access to the joint is made through two portals for inserting the arthroscopic camera and the surgical instruments. Existing solutions make use of instrumented touch probes for reconstructing bone surface and afterwards perform registration with a pre-operative model of the bone [21]. However, since the maneuverability inside the joint is limited, this procedure is often difficult. Also, existing image-based surface reconstruction procedures fail in providing acceptable results due to the very low texture of the bone surface. As future work, we intend to explore the possibility applying the new 3-DoF tracker to the arthroscopic images for the reconstruction of bone surface, which would then enable registration with the pre-operative model to be performed with schemes that make use of surface normals [19]. Additionally, we will further investigate how to perform multi-view photoconsistency refinement/tracking using the 3-DoF formulation in an efficient manner.

Acknowledgments. This work was funded by the Portuguese Science Foundation and COMPETE2020 program through project VisArthro (ref.: PTDC/EEIAUT/3024/2014). This paper was also funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 766850.

References

1. Baker, S., Matthews, I.: Lucas-kanade 20 years on: a unifying framework. Int. J. Comput. Vision **56**, 221–255 (2004). https://doi.org/10.1023/B:VISI.0000011205.11775.fd
2. Barath, D., Eichhardt, I., Hajder, L.: Optimal multi-view surface normal estimation using affine correspondences. IEEE Trans. Image Process. **28**(7), 3301–3311 (2019). https://doi.org/10.1109/TIP.2019.2895542

3. Barath, D., Matas, J., Hajder, L.: Accurate closed-form estimation of local affine transformations consistent with the epipolar geometry, pp. 11.1-11.12 (2016). https://doi.org/10.5244/C.30.11
4. Barath, D., Molnar, J., Hajder, L.: Novel methods for estimating surface normals from affine transformations. In: Braz, J., et al. (eds.) VISIGRAPP 2015. CCIS, vol. 598, pp. 316–337. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29971-6_17
5. Benhimane, S., Malis, E.: Homography-based 2D visual tracking and servoing. Int. J. Robot. Res. **26**(7), 661–676 (2007). https://doi.org/10.1177/0278364907080252
6. Bentolila, J., Francos, J.M.: Conic epipolar constraints from affine correspondences. Comput. Vis. Image Underst. **122**, 105–114 (2014). https://doi.org/10.1016/j.cviu.2014.02.004. http://www.sciencedirect.com/science/article/pii/S1077314214000307
7. Eichhardt, I., Barath, D.: Optimal multi-view correction of local affine frames. In: BMVC (2019)
8. Eichhardt, I., Hajder, L.: Computer vision meets geometric modeling: multi-view reconstruction of surface points and normals using affine correspondences, October 2017. https://doi.org/10.1109/ICCVW.2017.286
9. Hajder, L., Barath, D.: Relative planar motion for vehicle-mounted cameras from a single affine correspondence, December 2019
10. Hartley, R.I., Sturm, P.: Triangulation. Comput. Vis. Image Underst. **68**(2), 146–157 (1997). https://doi.org/10.1006/cviu.1997.0547
11. Koser, K., Beder, C., Koch, R.: Conjugate rotation: parameterization and estimation from an affine feature correspondence. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp. 1–8, June 2008. https://doi.org/10.1109/CVPR.2008.4587796
12. Köser, K., Koch, R.: Differential spatial resection - pose estimation using a single local image feature. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5305, pp. 312–325. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88693-8_23
13. Mei, C., Benhimane, S., Malis, E., Rives, P.: Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors. IEEE-TRO **24**(6), 1352–1364 (2008)
14. Mikolajczyk, K., et al.: A comparison of affine region detectors. Int. J. Comput. Vis. **65**, 2005 (2005)
15. Minh, N.L., Hajder, L.: Affine transformation from fundamental matrix and two directions. In: VISIGRAPP (2020)
16. Perdoch, M., Matas, J., Chum, O.: Epipolar geometry from two correspondences. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 4, pp. 215–219, August 2006. https://doi.org/10.1109/ICPR.2006.497
17. Pritts, J., Kukelova, Z., Larsson, V., Chum, O.: Radially-distorted conjugate translations. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1993–2001 (2017)
18. Raposo, C., Barreto, J.P.: πMatch: monocular vSLAM and piecewise planar reconstruction using fast plane correspondences. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 380–395. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_23
19. Raposo, C., Barreto, J.P.: Using 2 point+normal sets for fast registration of point clouds with small overlap. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 5652–5658, May 2017. https://doi.org/10.1109/ICRA.2017.7989664

20. Raposo, C., Barreto, J.P.: Theory and pratice of structure-from-motion using affine correspondences. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
21. Raposo, C., et al.: Video-based computer aided arthroscopy for patient specific reconstruction of the anterior cruciate ligament. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) MICCAI 2018. LNCS, vol. 11073, pp. 125–133. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00937-3_15
22. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, June 2008. https://doi.org/10.1109/CVPR.2008.4587706
23. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D slam systems. In: IEEE-IROS, October 2012
24. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Found. Trends. Comput. Graph. Vis. **3**, 177–280 (2008)
25. Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/