Accurate Line-Based Relative Pose Estimation With Camera Matrices

PEIHONG YU\textsuperscript{1,2,3}, CEN WANG\textsuperscript{1}, ZHIRUI WANG\textsuperscript{4}, JINGYI YU\textsuperscript{1}, (Member, IEEE), AND LAURENT KNEIP\textsuperscript{1}, (Member, IEEE)

\textsuperscript{1}School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China
\textsuperscript{2}Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China
\textsuperscript{3}University of Chinese Academy of Sciences, Beijing 100049, China
\textsuperscript{4}Motovis Intelligent Technologies Ltd., Shanghai 201203, China

Corresponding author: Peihong Yu (yuph@shanghaitech.edu.cn)

This work was supported in part by the Natural Science Foundation of Shanghai under Grant 19ZR1434000, in part by the National Natural Science Foundation of China under Grant 61950410612, Grant 61976138, and Grant 61977047, and in part by the Science and Technology Commission of Shanghai (STCSM) Program under Grant 17XD1402900.

ABSTRACT While most monocular structure-from-motion frameworks rely on sparse keypoints, it has long been acknowledged that lines represent an alternative, higher-order feature with high accuracy, repeatability, and abundant availability in man-made environments. Its exclusive use, however, is severely complicated by its inability to resolve the common bootstrapping scenario of two-view geometry. Even with stereo cameras, a one-dimensional disparity space, as well as ill-posed triangulations of horizontal lines make the realization of purely line-based tracking pipelines difficult. The present paper successfully leverages the redundancy in camera matrices to alleviate this shortcoming. We present a novel stereo trifocal tensor solver and extend it to the case of two camera matrix view-points. Our experiments demonstrate superior behavior with respect to both 2D-2D and 3D-3D alternatives. We furthermore outline the camera matrix’s ability to continuously and robustly bootstrap visual motion estimation pipelines via integration into a robust, purely line-based visual odometry pipeline. The result leads to state-of-the-art tracking accuracy comparable to what is achieved by point-based stereo or even dense depth camera alternatives.

INDEX TERMS Light-field cameras, multi-camera arrays, plenoptic vision, line features, trifocal tensor, automatic solver generator, Gröbner basis.

I. INTRODUCTION

Since the advent of sufficient computational resources in a compact format, visual localisation and mapping have become driving technologies in many applications that require a mobile system to automatically map and localize within GPS-denied, previously unknown environments. Important examples are given by robotics and virtual, augmented, or mixed reality. The present paper looks at solving this problem with regular cameras that simply perceive photometric projections of the environment. While popular alternatives have emerged in the form of active depth cameras, the regular passive case remains attractive due to a small form factor, no range limitations, and low energy consumption, which is why they already form part of existing products on the market. Despite the discovery of dense photometric registration methods [1], the state-of-the-art solutions for passive visual sensors currently remain sparse least-squares approaches that predominantly rely on feature extraction and bundle adjustment [2], [3]. The present paper explores the potential of passive multi-camera arrays, analyses the differences between the various camera array configurations, and presents a new algorithm for relative camera array pose estimation.

Our work is concerned with the type of features that are extracted from the images in such systems and mapped into a 3D representation. It has long been recognized that the addition of edge or line features can greatly contribute to the overall accuracy of a method, particularly in man-made environments that present lots of homogeneously colored surfaces with reduced texture. However, exclusive reliance on line features remains a challenge, as it turns the computation of the relative pose between two monocular images into an ill-posed problem. The back-projection of a 2D line
measurement is a plane that intersects with the camera center. For any 2D-2D line correspondence between two views, these planes will generally intersect in a 3D line, independently of the relative camera pose. Line-based solutions therefore require corresponding measurements between at least three views, the relative pose of which can then be recovered using trifocal tensor geometry [4]. Besides other challenges such as partial detections and robust line description, the difficulty of stably computing the trifocal tensor has since ever prevented the realization of a truly convincing, purely line-based, monocular visual odometry pipeline. An alternative is given by using a stereo camera, triangulating lines in each stereo view, and then aligning them using a 3D-3D registration paradigm. This is however fragile, as reduced baselines and vanishing disparities for lines that intersect with the epipole result in inaccurate 3D line locations.

We present the first purely line-based visual odometry solution that reliably solves common benchmark cases. This is achieved through the following three contributions:

- We set a new state-of-the-art for line-based stereo relative pose. The solver does not rely on inaccurate triangulations, but solves the problem directly from the original, measured 2D-2D line correspondences. The derivation starts from classical trifocal tensor geometry, and adapts it to the case of two stereo view-points.
- We demonstrate the applicability of this approach to the plenoptic case, and realize a powerful, purely line-based relative pose solver for camera matrices.\(^1\) We carefully exploit the redundancy in the views in order to benefit from as many as possible lines, however taking observability of depth given by their direction into account.
- In order to evaluate its potential for continuous tracking over real images, we embed our solver into a purely line-based visual odometry pipeline for camera matrices by adding line-based bundle adjustment in the back-end.

\(^1\)We define a camera matrix to be a square grid arrangement of perspective cameras. It may also be described as a light-field camera or multi-camera array.

The result demonstrates highly reliable and accurate, purely line-based motion estimation. The effectiveness of our approach is supported by our experimental results. Our purely line-based 2D-2D relative pose solver consistently outperforms existing alternatives, and our camera-matrix visual odometry framework is able to compete with some of the most recent RGBD-SLAM solutions in terms of tracking accuracy. Section III starts with the theory behind our trifocal tensor based solver and its application to camera matrices. Section IV outlines its inclusion into a visual odometry framework. Section V then presents our experimental results, where we apply our novel relative pose solver to both simulated and real data.

II. RELATED WORK

Line-based structure from motion has a long-standing history in the geometric computer vision community. Even early works such as [5] already demonstrate the feasibility of optimizing poses and structure by pure reliance on line features. [4] furthermore analyses the minimal case, and introduces the trifocal tensor to describe all scene independent relations in the minimal case (i.e. a triplet of views). The theory has later been summarized in [6]. Nearly a decade later, [7] and [8] introduce further seminal contributions detailing the representation, initialization, and optimization of 3D lines, possibly taking geometric priors into account.

While certainly providing a fundamental understanding of the geometry of lines, these contributions do not go as far as to provide the necessary cornerstone for a successful realization of a complete, purely line-based online structure-from-motion pipeline, which is a stable relative pose solver. The challenge of solely relying on lines has therefore been mostly circumvented by relying simultaneously on points and lines. Important early examples are given by [9] and [10], which introduce edge landmarks and an extended Kalman filter that estimates both points and lines, respectively. Reference [11] introduces a least-squares optimization-based alternative. More recently, we have experienced a revival of hybrid feature-based methods, with [12]–[15] providing monocular solutions, [16] a solution for RGB-D cameras, and [17], [18] alternatives for stereo cameras. Reference [19] proposes an extension to a popular direct point-based method [20] by utilizing lines to guide keypoint selection rather than to directly act as features. Reference [21] shows another possible combination of both modalities, where an initial point-based reconstruction is augmented through the addition of lines in a final back-end optimization step.

Despite its difficulty, there have been a few efforts on realizing purely line-based online structure-from-motion pipelines for photometric sensors [22]–[24]. Their results are however not very encouraging as [22] states that From the [...] results, there is no incentive to believe that lines constrain the motion sufficiently to be used alone, and [23] and [24] only present small scale experiments without evaluating motion accuracy at all. Further notable work has been presented in [25], which estimates Plücker line
coordinates through a Kalman filter. However, they again show only very few encouraging results, and—while demonstrating good empirical convergence properties—their filter-based bootstrapping does not provide any convergence guarantees. Reference [26] presents another interesting filtering-based solution, but circumvents the initialization question by utilizing prior information about the environment.

Some works achieved a purely line-based solution by exploiting the structural regularity of manmade environments. Reference [27] adopts Manhattan World constraints by simultaneously grouping and estimating vanishing directions, and then obtains both intrinsic parameters as well as absolute orientation by extracting the normalized vanishing points and enforcing their orthogonality. Given a triplet of lines of which two are parallel and the third is perpendicular to the others, [28] estimates the relative rotation from vanishing point information and the relative translation using the potential intersection points between the lines. Reference [29] adopts both vertical and floor lines, and obtains camera poses using an EKF framework. Reference [30] extends [29]'s work into a stereo graph-SLAM system, in which they use Plücker line coordinates for initialization and orthonormal representation during optimization. Reference [31] again estimates relative rotation from vanishing directions. They furthermore introduce a novel line parameterization and an extended Kalman filter for continuous camera pose estimation. Reference [32] does not strictly depend on the Manhattan World assumption, but still requires the presence of multiple parallel lines for several dominant directions in the scene. The method proceeds by clustering parallel lines and obtaining their 3D line direction, which are then aligned for relative rotation computation. The relative translation is again estimated by the method presented in [28]. While the assumptions of orthogonality, verticality, or planarity of lines are common, they constrain the derived algorithms to special environments only.

Another category of works that are naturally related to ours is given by [33], and [34]. They look at the classical, point-based structure-from-motion problem with light-field cameras. [35] furthermore exploits the ray-manifold structure of point and line correspondences inside a light-field frame in order to triangulate them in 3D. Unfortunately, the image-based geometric relations presented in [33]–[35] depend on ideal multi-camera arrays where all sensors are in a perfect plane and have the exact same orientation. Therefore, their application in the real world is limited to small scenes captured by high-accuracy light-field cameras, such as Lytro or Raytrix, and could not generalize well to the larger scenes. Finally, [36], [37], and [38] represent interesting attempts to realize online structure-from-motion (i.e. visual SLAM) pipelines. Despite promising results, they rely on points, and are only vaguely compared against the state-of-the-art set by the visual SLAM community.

In conclusion—by employing a camera matrix and leveraging trifocal tensor geometry—we present a novel, line-based relative pose solver that enables the first convincing, purely line-based visual odometry solution. It rivals the state-of-the-art given by point-based methods for normal cameras, or even direct RGBD methods.

### III. RELATIVE POSE FOR CAMERA MATRICES WITH LINES

As stated in [6], the trifocal tensor encapsulates all the (projective) geometric relations between three views that are independent of scene structure. Since two multi-perspective views (stereo or plenoptic) naturally provide more than three perspective views of a scene, the trifocal tensor relations seemingly represent a good starting point for constraining the unknown relative pose from lines. After a short review, we see how they can be transferred to first stereo, and then plenoptic cameras. The section concludes with the automatic generation of the actual pose solver.

#### A. A SHORT REVIEW OF TRIFOCAL TENSOR GEOMETRY

Suppose a line \( L \) in 3D space is imaged as the corresponding triplet \( m \leftrightarrow m' \leftrightarrow m'' \) across three perspective views, which are indicated in Figure 2(a) by their centers, \( C, C', \) and \( C'' \). Trifocal tensor geometry describes the existence of three \( 3 \times 3 \) matrices \( S \) (or alternatively a cubic tensor of size 3) which—if correctly factored with the lines in the second and the third view—permit the reconstruction of the three coordinates of the line in the first view, respectively. Calling the result \( q \), this line transfer is given by

\[
q^T = m'^T [S_1, S_2, S_3] m''
\]

(1)

Since \( q \) ideally has to be proportional to the measured line in the first view, \( m \), we obtain the incidence relation

\[
(m'^T [S_1, S_2, S_3] m'') m |x = 0^T,
\]

(2)

in which

\[
| m | = \begin{bmatrix}
0 & -m_3 & m_2 \\
m_3 & 0 & -m_1 \\
-m_2 & m_1 & 0
\end{bmatrix}
\]

The trifocal tensor can be described as a function of the three camera matrices \( P, P', \) and \( P'' \). Before proceeding, we first move to normalised coordinates given that we assume to be in the calibrated case. We start with normalising the line measurements \( m \) that satisfy the equation \( x^T m = 0 \) for any image point \( x \) along the line. Inserting the matrix of intrinsic camera parameters \( K \), we obtain

\[
x^T m = 0 \implies x^T K^{-T} K^T m = 0 \implies (K^{-1} x)^T (K^T m) = 0.
\]

(3)

The normalized line coordinates are therefore given by \( l = K^T m \), which is a 3-dimensional vector. \( f = K^{-1} x \) corresponds to a normalized line point and therefore a direction vector pointing towards the corresponding 3D point of \( L \). \( f^T l = 0 \), and \( l \) therefore also corresponds to the normal vector of the 3D plane spanned by the 3D line \( L \) and the camera.
Multiple-view geometry of 3D line observations.

FIGURE 2. Multiple-view geometry of 3D line observations.

Center C. Referring to the first view, \( P, P', \) and \( P'' \) in the calibrated case are

\[
P = [I | 0], \quad P' = [R' | t'], \quad \text{and} \quad P'' = [R'' | t''],
\]

where \( t', R' = [r_1', r_2', r_3'] \) and \( t'', R'' = [r_1'', r_2'', r_3''] \) enable the euclidean transformation of points from the first to the second and the first to the third camera frame. Given its general form taken from [6], the trifocal tensor \( [T_1, T_2, T_3] \) that satisfies the incidence relation in the calibrated case

\[
(t^T [T_1, T_2, T_3] t'') | I \times = 0 \quad (4)
\]
is finally given by

\[
T_i = r' t''^T - t' r_i''^T. \quad (5)
\]

For further details on the trifocal tensor including its derivation, the reader is kindly referred to Chapter 15 of [6].

B. LINE-BASED STEREO RELATIVE POSE

Let us assume the existence of a calibrated stereo rig for which intra-camera rotations are known and compensated for (e.g. by applying the homography at infinity). The extrinsic stereo calibration parameters are therefore given by \( e_1 \) and \( e_2 \), which denote the translational displacements of the left and right camera centres from a common stereo reference frame. Let us furthermore describe the relative pose between two subsequent stereo frames by the parameters \( \{t, R\} \), which transform points from the first to the second stereo frame. The geometry is illustrated in Figure 2(b).

The application of trifocal tensor geometry to the line-based stereo relative pose problem is straightforward. Any three distinct views selected from the two stereo frames need to satisfy the trifocal tensor incidence relation (4). Picking the left view of the first stereo frame to be the first view of our trifocal set, the right view of the second stereo frame to be the third view of the trifocal set, we can easily see that

\[
R' = I, t' = e_1 - e_2, R'' = R \text{ and } t'' = t + Re_{1} - e_2. \quad (6)
\]

Substituting these expressions in (5) and (4), we obtain three constraints on the relative pose parameters between the two stereo frames. We can easily obtain three more equations by choosing the right view of the second stereo frame as the third view in our trifocal set, which will change \( t'' = t + Re_1 - e_2 \). It is furthermore intuitively clear that a single line correspondence across two stereo views is not sufficient to fully constrain the relative pose. A second line correspondence is needed, from which we can obtain another 6 constraints on the relative pose. With a total of 12 equations that are linear in the parameters \( t \) and \( R \), we could already obtain a first solver by a simple linear solution. This has in fact been done in [39], however concluding that solutions through linearization are unstable under noise and unable to resolve solution multiplicities. Reference [39] also presents a non-linear solver to the stereo line-based relative pose problem that implicitly enforces all constraints of orthonormality on \( R \), however residing to a manually constructed, unbalanced variable elimination strategy. Besides an extension to camera matrices, the remainder of this section establishes a novel higher-order polynomial solver. It works for an arbitrary number of line correspondences and relies on the more modern strategy of solving the first-order optimality conditions with the Gröbner basis method.

C. EXTENSION TO CAMERA MATRICES

A camera-matrix frame can be regarded as the union of many perspective images that are captured with small baselines in between. The focus of the present paper lies on the standard configuration of square, planar camera matrices as illustrated in Figure 1. The availability of many views naturally raises the question which trifocal sets to choose among all available ones. In the present paper, we use \( 3 \times 3 \) matrices of cameras, which would lead to \( 9 \times 8 \times 9 = 648 \) possible trifocal triplets, and a large number of possible intra and inter camera-matrix matching procedures. The problem however becomes more manageable if considering the fact that not all line correspondences are equally useful. For example, any 3D line that is parallel to or intersects with the baseline between two views will receive the same coordinates upon reprojection into those two views, and would thus be impossible to reconstruct in 3D or even use for constraining the relative pose. Seen from the image plane, these are lines that intersect with the epipole. Furthermore, in analogy to points, a wider baseline generally causes larger line displacements and thus a better signal-to-noise ratio for deriving the relative pose.

These insights can be leveraged to perform a smart selection of trifocal sets. Starting with the two views taken from 2Note that there are of course two more trifocal triplets that can be chosen among the four frames, but it is easily verified that they are no longer linear in \( t \) and \( R \).
cases are illustrated in Figure 3. For every new camera-matrix on the relative pose via two trifocal sets. The restricted line each camera-matrix frame), from which we obtain constraints ing line correspondences across 4 views (one stereo view in significant changes. Our strategy therefore consists of form-

base on the assumption that the relative rotation is small,

the second camera-matrix frame for forming trifocal sets 

will generally work.

D. SOLVER GENERATION

the first camera-matrix frame, we only consider pairs of 

views with opposite baselines, thus ensuring maximal dis-
placements of the line observations. In order to furthermore ignore lines which are too close to the epipoles (which are 

known in advance), we remove any lines for which the absolute value of the sine of the angle between its direction vector and the direction towards the epipole (at infinity) is too small. Lastly, we only consider the same two views in the second camera-matrix frame for forming trifocal sets based on the assumption that the relative rotation is small, and that—as a result—line directions do not undergo any significant changes. Our strategy therefore consists of form-

ing line correspondences across 4 views (one stereo view in each camera-matrix frame), from which we obtain constraints on the relative pose via two trifocal sets. The restricted line directions, matching steps, and trifocal sets for two such cases are illustrated in Figure 3. For every new camera-matrix frame, we perform 4 intra-frame matching steps for the stereo pairs, and four inter-frame matching steps to match the first view of each virtual stereo pair to its corresponding subframe in the previous matrix frame.

where $n$ indicates the total number of the trifocal sets. We eliminate the translation parameters, which simply appear linearly in the equations. We have

$$
t = -(A_i^T A_1)^{-1} A_i^T A_2 \overrightarrow{r} - (A_i^T A_1)^{-1} A_i^T a_3. 
$$

Proposition: $A_i^T A_1$ has full rank.

Proof: We have

$$
A_i^T A_1 = \sum_{i=1}^{n} A_i^T A_{1i} = \sum_{i=1}^{n} I_i'(d_i \times I_i')^T (d_i \times I_i')^T 
$$

$$
= \sum_{i=1}^{n} s_i l_i'I_i'T_i = \sum_{i=1}^{n} s_i l_i'I_i'T_i 
$$

which are three linear constraints on the variables $t$ and $R$. The exact form of matrices $A_{1i}$, $A_{2i}$, and vector $a_{3i}$ is presented in Appendix A.

Two lines are sufficient to constrain the relative pose, and—given stereo observations inside each camera-matrix frame—they lead to 4 trifocal sets and thus 4 groups of 3 linear constraints. We start by choosing an alternative, minimal parametrization of the relative rotation, namely Cayley parameters [40]. While this renders the equations non-linear in the rotation, it has the advantage of explicitly enforcing the required orthonormality. The equations are solvable using polynomial elimination theory, and an application of the latter immediately reveals that the 12 equations formed from two stereo line correspondences are in fact already more than required. A subset of 8 equations would have to be chosen in order to form a minimal case. This leaves us with the difficult question which ones to choose.

Since two distinct lines remain a requirement, we circumvent this question by changing our problem formulation. We notably construct the sum of squared algebraic errors resulting from each incidence relation, and generate a new minimal solver that seeks all stationary points of this energy, in closed-form. While this solver is slightly more expensive, the alternative underlying problem has the great advantage of making full use of 2 stereo line correspondences. More importantly, the energy can be integrated over any number of stereo line correspondences greater or equal to 2, the solver will generally work.

We start by stacking the algebraic constraints for each trifocal set, leading to

$$
\begin{bmatrix}
A_{11} \\
\vdots \\
A_{1n}
\end{bmatrix}
\begin{bmatrix}
t \\
\vdots \\
a_{3n}
\end{bmatrix}
+ \begin{bmatrix}
A_{21} \\
\vdots \\
A_{2n}
\end{bmatrix}
\overrightarrow{r}
+ \begin{bmatrix}
a_{31} \\
\vdots \\
a_{3n}
\end{bmatrix}
= A_1 t + A_2 \overrightarrow{r} + a_3 = 0,
$$

where $n$ indicates the total number of the trifocal sets. We eliminate the translation parameters, which simply appear linearly in the equations. We have

$$
t = -(A_i^T A_1)^{-1} A_i^T A_2 \overrightarrow{r} - (A_i^T A_1)^{-1} A_i^T a_3. 
$$

Note that, while the Cayley parametrization is not free of singularities, the relative pose problem often has a limited rotation angle. The use of the minimal Cayley parametrization therefore remains beneficial, as—unlike with quaternions—it does not induce symmetries in the problem, and avoids the introduction of additional unit norm constraints.
which implies that $A_1^T A_1$ is positive definite and has full rank. In particular, it is easy to see that if the $v_i^n$ originate from two non-parallel lines that are constructed using different trifocal tensors, they will span a full rank space, thus fulfilling the above condition. It is also clear that having non-parallel lines is a condition for our solver to work, as it prevents unobservable parallax by sliding the camera-matrix frames in the direction parallel to the lines. We do not compute solutions for any pair of lines that are too parallel. $A_1^T A_1$ therefore automatically stays invertible for any correspondences that we evaluate. □

Back substituting (9) in (8), we obtain

$$A^* \mathbf{T} a^* = 0,$$

where

$$\begin{align*}
A^* &= A_2 - A_1 (A_1^T A_1)^{-1} A_1^T A_2 \\
a^* &= a_3 - A_1 (A_1^T A_1)^{-1} A_1^T a_3.
\end{align*}$$

(12)

We now turn our attention to the minimal Cayley parametrization [40] of $R$, which is given by (13), as shown at the bottom of this page, where $c = 1 + x^2 + y^2 + z^2$.

We first substitute (13) into (12). Then using $\mathbf{q} = [q_1^T q_2^T q_3^T]^T$, (12) is transformed into

$$A^* \mathbf{q} + a^* c = M^* s = 0,$$

where $s = [x^2, yx, y^2, xz, yz, z^2, x, y, z]^T$ forms all monomials of the Cayley parameters $\{x, y, z\}$. The detailed form of $M^*$ is indicated in Appendix B. We can finally define the squared scalar residual over all measurements

$$E = s^T M^* T M^* s = s^T M s,$$

(15)

and compute all solutions for $\{x, y, z\}$ from the first order optimality conditions $2 s^T M \left[ \frac{\partial}{\partial x} \frac{\partial}{\partial y} \frac{\partial}{\partial z} \right] = 0$. We create a solver for this problem from random experiments. Since the norm of the line measurements does not matter for the validity of (4), they can be constructed from random 3D end-points and relative camera poses defined over a finite prime field. After deriving the optimality conditions in the finite field, we apply the automatic solver generator of [41]. The size of the produced elimination template is $92 \times 119$, and the corresponding action matrix has a size of 27 (which also equals to the maximum number of stationary points in the problem). The action matrix is to be understood as a multi-variate extension of the companion matrix. It is composed by the coefficients of the Gröbner basis, and permits the extraction of the solutions by eigen-decomposition. For a detailed review of the Gröbner basis theory and its application in automatic closed-form solver generation, we kindly refer the reader to [41], [43]. Once the rotation is retrieved, the translation can be found by back-substitution into (9). Our code is in the supplemental material and will be published.

IV. LINE-BASED VISUAL ODOMETRY FOR CAMERA MATRICES

We embed our novel line-based relative pose solver into an online visual odometry pipeline. We use state-of-the-art algorithms — EDLines [44] to detect line segments in each image, and LBD descriptors [45] to match them between sub-sequent frames (8 intra and inter frame stereo matching procedures as outlined in Section III-C). Intra-frame matching is furthermore supported by geometric cues to address any remaining issues such as fragmented lines or mismatches. An overview of the entire framework is given in Figure 4. It consists of a front-end and a back-end module, which perform visual odometry and local mapping, respectively. Both modules are detailed in the following.

A. VISUAL ODOMETRY FRONT-END

After all potential correspondences over trifocal triplets of views are extracted, we apply our solver within RANSAC [46] to robustly identify inliers and initialize the relative pose of the new camera-matrix frame. In each iteration, we utilize the four trifocal triplets that can be extracted from two line measurements across two stereo sub-sets. The success of the relative pose solver obviously depends on the angle between the considered 3D lines. We therefore immediately discard random samples for which the directions of the line measurements in the image are too similar. We furthermore discard rotations which are too far from identity. Once RANSAC terminates, we run a final iteration of the solver over all inlier correspondences. In order to classify inliers, we evaluate a uni-directional transfer error. We use the hypothesized trifocal tensor to transfer the line measurement from the second and third view back to the first one, and then evaluate the orthogonal distance of the end-points of the original line measurement in the first view. However, in order to preserve longer lines for which this error is much more sensitive w.r.t. pose errors, the transfer error is scaled down by the euclidean distance between the end-points in the image.

B. LOCAL MAPPING BACK-END

Our local mapping module performs optimization of all variables over a window of $k$ most recent keyframes. A new

$$R = \frac{1}{c} \cdot Q = \frac{1}{c} \cdot [q_1^T q_2^T q_3^T]$$

$$= \frac{1}{c} \cdot \begin{bmatrix}
1 + x^2 - y^2 - z^2 & 2xy - 2z & 2xz + 2y \\
2xy + 2z & 1 - x^2 + y^2 - z^2 & 2yz - 2x \\
2xz - 2y & 2yz + 2x & 1 - x^2 - y^2 + z^2
\end{bmatrix}.$$  

(13)
keyframe is added when its relative rotation with respect to the previous keyframe is above 5 degrees or the length of its relative translation is above a scene depth-related threshold. The number of keyframes is fixed to 4 in our experiments, which empirically optimizes the trade-off between computational efficiency and the density of the local view-graph.

The optimization includes the poses of the camera-matrix frames as well as all parameters of 3D lines for which there exist at least two observations inside the window of considered frames. The procedure is commonly known as local or windowed bundle adjustment. After a first execution terminates, we run a map matching procedure in which we seek further correspondences between the current frame and the 3D lines, directly. After all correspondences have been added, we conclude with another round of local bundle adjustment. With better constraints provided by the map matching step, the estimation accuracy is further improved.

An important question when applying bundle adjustment is how exactly to parametrize the optimized variables. Taking the camera pose in the calibrated case as an example, the difficulty lies in making sure that the orientation variable remains a point on the special-orthogonal group $SO(3)$, either through a minimal, implicit parametrization, or by explicitly adding side-constraints to the optimization. In the spirit of consistency, we simply choose the minimal Cayley parameters [40] to represent the orientation of each camera-matrix frame. Furthermore, in order to ensure that we remain far away from $180^\circ$ rotations for which $\| [x \ y \ z] \| \to \infty$, we optimize pose changes from the initial absolute pose of each camera-matrix frame rather than the absolute pose directly.

We also need to decide how to parametrize the 3D lines, which are commonly represented by 6 Plücker line coordinates, but only have 4 degrees of freedom. The Plücker line coordinates are notably given by a line direction vector $d$ with unit norm and a moment vector $m$ that is orthogonal to $d$. As suggested in [6], a minimal parametrization could be given by two points on two planes, which would however leave us with the difficulty of defining these two planes as well as the inability to attain lines that are parallel to one of the two planes without being contained in it. With strong ties to [7], we propose a less restrictive minimal parametrization of the 3D lines for back-end optimization. Since $d$ and $m$ are orthogonal, they can notably be parametrized as the first two columns of a rotation matrix that again is represented as a function of Cayley parameters [40]. The fact that the moment vector $m$ does not have unit norm is furthermore reflected by multiplying the second column of this matrix by a scale parameter $s$. We obtain

$$[d, m] = \frac{1}{c} \begin{bmatrix} 1 + x^2 - y^2 - z^2 \\ 2xy + 2z \\ 2xz - 2y \end{bmatrix} s \begin{bmatrix} 2xy - 2z \\ 2yz + 2x \\ 1 - x^2 + y^2 - z^2 \end{bmatrix},$$

(16)

where $c = 1 + x^2 + y^2 + z^2$. We furthermore switch to local optimization by adding a starting orientation $R_0 = \begin{bmatrix} d_0 \ |m_0| \\ (d_0 \times m_0) \end{bmatrix}$ and a starting line moment $s_0$ to the parametrization, thus resulting in

$$[d, m] = R_0 \frac{1}{c} \begin{bmatrix} 1 + x^2 - y^2 - z^2 \\ 2xy + 2z \\ 2xz - 2y \end{bmatrix} (s_0 + s) \begin{bmatrix} 2xy - 2z \\ 2yz + 2x \\ 1 - x^2 + y^2 - z^2 \end{bmatrix}. \quad (17)$$

As required, this parametrization has only 4 degrees of freedom, and implicitly enforces all side-constraints on the Plücker line coordinates.

We conclude the exposition of our line-based bundle adjustment back-end by explaining the reprojection error of every 3D line into every camera-matrix sub-frame where a measurement is available. Measurements are notably given by two end points in each respective sub-frame, while our
3D lines are parametrized as infinitely long. The reprojection error is simply given by the sum of the two squared orthogonal distances between the end points and the reprojected line. Note that, in contrast to [12], [17], we employ a genuine, virtually infinite line representation in 3D rather than two end-points. In turn, we utilize end-points where they naturally occur, namely for our measurements extracted in the image plane. This appears to be a more logical definition, as end-points defined in 3D could easily collapse and produce a zero residual.

V. RESULTS

Our experimental evaluation splits into two parts. We first analyze our plenoptic, trifocal tensor based 2D-2D registration on various simulated cases, and compare against the more intuitive 3D-3D registration alternative. We then analyze its performance for line-based visual odometry, and compare against state-of-the-art visual tracking solutions on a common benchmark sequence. We conclude with a demonstration on a real system.

A. VALIDATION OF THE TRIFOCAL-TENSOR BASED SOLVER

We first validate our line-based 2D-2D registration method on simulated experiments, and compare against a simple 3D-3D registration alternative that is preceded by triangulation in each frame. A basic simulation experiment is generated as follows. We hypothesize a random relative rotation and translation for the second camera-matrix frame, as well as random 3D end-points for two lines constrained to lie within the volume defined by $x \in [-3, 3], y \in [-2, 2], z \in [3, 5]$ (i.e. in front of the first camera-matrix frame), the size of which is chosen according to a realistic distribution of lines in regular indoor scenes. We then project the 3D points onto all image planes assuming all intrinsic parameters are equal and given by a focal length of 480 and a principal point of [320 240] T, and there are 9 camera centres arranged in a $3 \times 3$ grid of 0.5 m width and height. We finally add random uniform noise of at most 0.5 pixels to all image coordinates. For the 3D-3D registration method, we first use the state-of-the-art triangulation method outlined in [35], an accurate and robust algorithm that works under light-field/camera-matrix settings (i.e. the configuration we address), to recover noise-affected 3D line measurements in each camera-matrix frame, and then align them by first recovering the relative rotation from the line directions, and then the relative translation from a straightforward linear equation.

We evaluate the solvers from seven different aspects, which are precision for 1) varying rotation angles, 2) varying...
translation lengths, 3) varying noise levels for the line endpoints, 4) different numbers of line correspondences, 5) the angle between the lines in the minimal case of 2 lines, 6) the length of the lines, and 7) the intra-camera-matrix baseline. For each experiment, we only change one of the parameters and leave all others unchanged with respect to the basic experiment configuration outlined above. Detailed parameter variations are indicated in Table 1. We run 5000 independent runs for each experiment. The mean and median errors are finally reported in Figure 5. As can be observed, our direct 2D-2D registration method consistently outperforms the 3D-3D registration alternative. It proves that the error in the triangulated lines has a negative impact on the 3D-3D registration alternative. We confirm that the success of online visual motion estimation depends crucially on a triangulation-independent registration method.

We also include the linear solver from [39] using \{3 lines\} into our experiments. We run 5000 independent runs for each method with the default setting of the six parameters. Histograms representing distributions over errors are shown in Figure 6, which indicates that our method has more runs with smaller errors, and thus outperforms the linear solver in both robustness and accuracy.

### TABLE 2. Median absolute trajectory errors averaged over 50 runs on ICL-NUIM sequences $kt_1$ and $kt_2$. Unit: cm.

| Sequence | Out. (0.1m baseline) | ORB-SLAM 2 (F) | ORB-SLAM 2 (R) | Essential Pose (47) | DVO-SLAM (48) | ENDOR et al. (49) | MRSMap [50] | KittiRus [51] |
|----------|---------------------|----------------|----------------|---------------------|---------------|----------------|-------------|-------------|
| $kt_1$   | 1.9                 | 8.6            | 17.1           | 0.9                 | 2.9           | 0.8            | 22.8        | 0.5         |
| $kt_2$   | 0.7                 | 6.8            | 1.9            | 1.4                 | 1.8           | 1.8            | 18.9        | 1.0         |

### TABLE 3. Detailed results for our method on ICL-NUIM sequences $kt_1$ and $kt_2$.

| Baseline [m] | \(\text{ATE} \text{ RMSE} \text{ (cm)}\) | \(\text{RPE trans. RMSE} \text{ (cm)}\) | \(\text{RPE rot. RMSE} \text{ (deg)}\) | \(\text{ATE} \text{ RMSE} \text{ (cm)}\) | \(\text{RPE trans. RMSE} \text{ (cm)}\) | \(\text{RPE rot. RMSE} \text{ (deg)}\) | \(\text{ATE} \text{ RMSE} \text{ (cm)}\) | \(\text{RPE trans. RMSE} \text{ (cm)}\) | \(\text{RPE rot. RMSE} \text{ (deg)}\) |
|--------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| \(0.05\)     | 3.76                            | 3.08                            | 10.41                           | 0.51                            | 6.03                            | 3.50                            | 15.56                           | 0.76                            |
| \(0.1\)      | 1.29                            | 0.91                            | 3.32                            | 0.31                            | 4.57                            | 3.96                            | 7.67                            | 0.49                            |
| \(0.15\)     | 3.65                            | 3.24                            | 12.39                           | 1.48                            | 7.51                            | 6.06                            | 12.99                           | 3.71                            |
| \(0.2\)      | 5.93                            | 6.17                            | 8.74                            | 3.41                            | 10.91                           | 11.69                           | 18.97                           | 7.25                            |
| \(0.05\)     | 1.01                            | 0.72                            | 6.09                            | 0.38                            | 1.87                            | 1.38                            | 11.00                           | 0.64                            |
| \(0.1\)      | 0.85                            | 0.74                            | 3.54                            | 0.36                            | 1.55                            | 1.34                            | 6.03                            | 0.69                            |
| \(0.15\)     | 2.95                            | 2.71                            | 8.21                            | 1.00                            | 3.25                            | 3.08                            | 9.53                            | 0.79                            |
| \(0.2\)      | 8.24                            | 8.07                            | 15.36                           | 1.21                            | 8.24                            | 8.47                            | 15.36                           | 1.21                            |

### TABLE 4. Number of inlier line correspondences on ICL-NUIM sequence $kt_2$ regarding different baselines.

| Baseline [m] | Number of inlier line correspondences |
|--------------|--------------------------------------|
| 0.05         | \(60.11\)                            |
| 0.1          | \(47.74\)                            |
| 0.15         | \(39.57\)                            |
| 0.2          | \(33.33\)                            |
Drift is hardly observable, and the high accuracy is underlined by numbers. Although an RGB-D camera is a different type of sensor and thus not directly comparable, it is interesting to observe that we can achieve similar performance in terms of motion estimation accuracy despite only using passive photometric sensors. We also slightly outperform all comparison algorithms on sequence kt2. The sequence kt1 contains several frames in which only few lines can be detected, which limits the achievable ATE. Nonetheless, our algorithm maintains excellent performance with respect to state-of-the-art passive stereo alternatives. Further details about the results obtained by our method including the mean, median, maximum, and minimum errors of both the absolute trajectory error and the relative pose error can be found in Table 3. Even if changing the intra-camera-matrix baselines to a different value, sub-centimetre accuracy remains achievable. Note that when increasing the camera matrices’ baseline, the accuracy rises first and falls afterwards, which is against the observation in Figure 5. This issue is caused by the decreased number of available feature correspondences. In simulation, we can assume that the image plane is infinitely large, thus the cameras always observe all the features we add in virtual 3D space. However, real cameras have limited field of view, which causes the number of feature correspondences observed by the virtual stereo views to decrease as the baseline increases (see Table 4). Therefore, we confirm that there is only one optimal baseline for a given average scene depth.

C. COMPARISON AGAINST DIFFERENT SETUPS ON ICL-NUIM SEQUENCE kt2

We first run a stereo version of our own framework by disabling all but two of the camera-matrix sub-frames. We confirm that purely line-based stereo visual odometry with a horizontal intra-camera baseline is unable to produce reliable results, as horizontal lines become largely unusable. The remaining lines are often vertical, which still renders part of the motion unobservable. Since this setup hardly enables reliable motion estimation, we do not demonstrate its results. We instead run a second version of this experiment where we choose the top left and the bottom right views to form a stereo rig with a diagonal baseline. This arrangement successfully measures disparities in both horizontal and vertical directions, and hence is able to track the camera. However, as illustrated in Figure 7, the performance is still not satisfying, as multiple tracking failures occur. Upon tracking failure, the algorithm is simply restarted. The trajectory for this stereo camera setup is therefore composed of several parts.

We also evaluate our algorithm for multiple camera array configurations with different number of views and differently

![Figure 8. Eight cases of representative setups.](image)

| TABLE 5. Detailed results on ICL-NUIM sequence kt2 regarding different setups. |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|-------------------------------|
|                                | Error combined                  | Error full                      | Lost tracking                  |
|                                | mean | median | max  | min  | mean | median | max  | min  |               |
| Case 1  | ATE RMSE [cm] | 1.38 | 1.00  | 15.05 | 0.51 | 6.76 | 7.36  | 9.18 | 2.81 |
|        | RPE trans RMSE [cm] | 2.33 | 1.83  | 22.50 | 0.94 | 11.51 | 12.15 | 14.81 | 6.39 |
|        | RPE rot RMSE [deg] | 0.48 | 0.33  | 6.47  | 0.12 | 2.63 | 2.81  | 4.55 | 1.12 |
| Case 2  | ATE RMSE [cm] | 0.67 | 0.67  | 1.12  | 0.36 | 3.15 | 1.92  | 12.18 | 0.63 |
|        | RPE trans RMSE [cm] | 1.24 | 1.13  | 2.74  | 0.54 | 3.96 | 3.26  | 19.72 | 1.20 |
|        | RPE rot RMSE [deg] | 0.23 | 0.19  | 0.60  | 0.10 | 1.42 | 0.81  | 4.44 | 0.17 |
| Case 3  | ATE RMSE [cm] | 0.10 | 0.73  | 4.82  | 0.46 | 3.36 | 3.37  | 5.53 | 1.20 |
|        | RPE trans RMSE [cm] | 1.70 | 1.29  | 6.60  | 0.75 | 5.78 | 5.33  | 9.60 | 2.15 |
|        | RPE rot RMSE [deg] | 0.35 | 0.21  | 2.21  | 0.10 | 1.14 | 1.02  | 2.38 | 0.90 |
| Case 4  | ATE RMSE [cm] | 0.73 | 0.64  | 1.55  | 0.36 | 3.77 | 3.13  | 5.39 | 1.14 |
|        | RPE trans RMSE [cm] | 1.34 | 1.10  | 3.04  | 0.59 | 5.91 | 5.81  | 9.83 | 2.13 |
|        | RPE rot RMSE [deg] | 0.26 | 0.20  | 0.78  | 0.10 | 1.45 | 1.22  | 2.85 | 0.82 |
| Case 5  | ATE RMSE [cm] | 0.65 | 0.64  | 1.11  | 0.36 | 1.80 | 1.57  | 5.62 | 0.62 |
|        | RPE trans RMSE [cm] | 1.17 | 1.03  | 3.13  | 0.64 | 3.17 | 2.94  | 8.30 | 1.20 |
|        | RPE rot RMSE [deg] | 0.23 | 0.19  | 0.66  | 0.09 | 0.80 | 0.70  | 1.90 | 0.27 |
| Case 6  | ATE RMSE [cm] | 0.53 | 0.52  | 0.86  | 0.32 | 2.51 | 2.52  | 3.51 | 0.63 |
|        | RPE trans RMSE [cm] | 1.11 | 1.04  | 2.30  | 0.53 | 4.65 | 4.09  | 8.72 | 1.49 |
|        | RPE rot RMSE [deg] | 0.20 | 0.17  | 0.45  | 0.09 | 1.07 | 1.02  | 2.77 | 0.38 |
| Case 7  | ATE RMSE [cm] | 0.46 | 0.45  | 0.82  | 0.27 | 2.39 | 2.08  | 5.89 | 0.55 |
|        | RPE trans RMSE [cm] | 0.91 | 0.87  | 1.95  | 0.48 | 4.36 | 3.86  | 9.97 | 1.25 |
|        | RPE rot RMSE [deg] | 0.17 | 0.15  | 0.39  | 0.08 | 0.96 | 0.79  | 2.23 | 0.17 |
| Case 8  | ATE RMSE [cm] | -    | -     | -     | -    | 0.85 | 0.74  | 3.54 | 0.36 |
|        | RPE trans RMSE [cm] | -    | -     | -     | -    | 1.53 | 1.34  | 6.03 | 0.69 |
|        | RPE rot RMSE [deg] | -    | -     | -     | -    | 0.32 | 0.30  | 0.95 | 0.10 |
oriented baselines, which are shown in Figure 8. For some setups, tracking is prone to get lost because of insufficient views. We therefore evaluate 2 kinds of errors: “Error combined”, which is the combined error of all separate parts in cases with tracking failure, and “Error full”, which is the error in cases without tracking failure. We run 50 independent runs for each setup and the detailed results are summarized in Table 5. We can observe that the errors and the ratio of failure cases decrease as more views and therefore more virtual stereo baselines are added to the matrix. Furthermore, it can be observed that the direction of the baseline also matters. As line features are often horizontal or vertical, diagonal baselines lead to better results than horizontal/vertical baselines. Best results are achieved with configurations that have both diagonal and horizontal/vertical baselines.

Though case 7 has better accuracy than case 5 in “Error combined”, it seems surprising that it has worse accuracy in “Error Full”. A possible explanation is that since there are fewer failure cases for case 7, “Error full” for the final 6-view case is evaluated over more cases, and it may include some cases that are challenging possibly explaining the slightly lower accuracy. What can however be concluded is that more views mean better robustness.

D. APPLICATION TO REAL DATA

Most real-world stereo datasets are limited to regular stereo cameras and do not provide the additional views that we need for the evaluation of our camera-array-based framework. Furthermore, their baseline is mostly parallel to the ground, and—as shown in our experiments—this configuration is not advantageous for our line-based solver. We therefore underwent the effort of recording custom datasets.

In a final experiment, we demonstrate a successful application to real data collected with our own, custom-designed camera-matrix camera (cf. Figure 1). The camera rig is constructed with 9 GoPro HERO4 with baselines around 9 centimetres, which targets indoor scenarios. All the videos are collected with 1280 × 740 resolution and 60Hz frame rate, but are downsampled to 960 × 540 and 30Hz in our experiments. We pre-calibrate the camera system by taking images of a chessboard and using the method from [54], which calibrates both intrinsic and extrinsic parameters. The videos of the moving camera rig are furthermore time-synchronized by recording not only images, but also sound clicks throughout the dataset. Due to low ambient noise, these clicks are easily extracted from the soundtrack and aligned using simple cross-correlation in Matlab. Distortion and rotation between the different views in the matrix are finally compensated for by applying the calibration parameters. An example qualitative result is indicated in Figure 9, where we move the camera along an elliptic trajectory in the \( \{x, y\} \) plane. We also extract high-quality baselines by applying COLMAP [55], [56] and ORB-SLAM2 [3] (stereo), and observe little deviation by our method.

Using COLMAP’s results as the ground truth, we further include quantitative results for the eight cases of representative setups in Table 6. 50 independent runs are executed for each setup. The results are generally consistent with the previous observation in synthetic experiments. We obtain better accuracy and robustness when more views and therefore more virtual stereo baselines are added to the matrix.

E. PRACTICAL CONCERNS

Our framework is implemented in C++ and relies on OpenCV and the Ceres non-linear optimization toolbox. All experiments are conducted on an Intel Core i7 clocked at 3.60 GHz, and the implementation uses hyper-threading on all 4 cores in the OpenCV-based line extraction and matching and Ceres-based backend optimization parts of the implementation. For a 3 × 3 matrix of VGA cameras, our current implementation consumes an average of 86ms for line extraction, 8ms for intra-frame stereo matching, 3ms for inter-frame stereo matching, and 59ms for the robust rel-

| Case 1 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 23.10 | 24.93 | 33.84 | 10.92 |
| RPE rot. [deg] | 4.25 | 4.11 | 6.92 | 2.36 |

| Case 2 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 19.17 | 20.21 | 34.95 | 5.90 |
| RPE rot. [deg] | 4.41 | 5.86 | 8.32 | 2.45 |

| Case 3 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 18.51 | 19.12 | 33.03 | 8.29 |
| RPE rot. [deg] | 4.45 | 4.22 | 8.37 | 5.66 |

| Case 4 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 17.29 | 18.62 | 28.09 | 8.35 |
| RPE rot. [deg] | 5.10 | 5.11 | 7.37 | 3.87 |

| Case 5 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 16.50 | 16.01 | 35.88 | 8.78 |
| RPE rot. [deg] | 3.56 | 3.51 | 6.26 | 2.03 |

| Case 6 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 15.67 | 14.92 | 24.48 | 9.31 |
| RPE rot. [deg] | 3.90 | 3.64 | 6.66 | 2.03 |

| Case 7 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 11.06 | 10.88 | 20.01 | 6.11 |
| RPE rot. [deg] | 3.79 | 3.90 | 5.89 | 2.41 |

| Case 8 | ATE [cm] | med | min | max |
|--------|----------|-----|-----|-----|
| RPE trans. [cm] | 13.66 | 12.38 | 23.39 | 8.03 |
| RPE rot. [deg] | 3.66 | 2.85 | 5.47 | 1.91 |

FIGURE 9. An example result from our real data experiments.
ative pose computation, all on a standard CPU architecture. The mapping part consumes an average of 7ms for line triangulation, 90ms for map matching, and 75ms for local bundle adjustment. Note however that mapping is only executed whenever a new keyframe is defined, and can furthermore be executed in a parallel thread. We therefore achieve an overall frame-rate of about 6Hz. While there is certainly room for improvement (notably through parallelisation), our implementation is able to process slow motion in real-time.

VI. DISCUSSION

We present a novel stereo trifocal tensor solver that achieves state-of-the-art accuracy, and successfully embed it into an algorithm to compute the motion of a camera matrix. The result is the first visual odometry framework for camera matrices that achieves state-of-the-art performance on common benchmark sequences. Comparison algorithms include both passive and active, feature-based and direct methods. What is perhaps most surprising is that this result has been achieved by pure reliance on line features. We successfully demonstrate that—in contrast to monocular or stereo setups—the multiple, differently oriented baselines of a camera matrix lead to the necessary level of robustness to handle the purely line-based scenario, eventually resulting in the outstanding tracking accuracy that one would expect from regressed line measurements. We consider this an exciting result that motivates further exploitation of plenoptic vision for an economic realization of next-generation visual localisation and mapping solutions.

APPENDIX A

Here we present the form of matrices $A_{11}, A_{21},$ and $a_{3i}$. Let $\mathbf{r} = [r_1^T, r_2^T, r_3^T]^T$. Rearranging $I_{proj} = I^T [T_1, T_2, T_3] I'$, we get the following equation

$$I_{proj} = \begin{bmatrix} a_{11} I + a_{12} \mathbf{r} + a_{13} \\ a_{21} I + a_{22} \mathbf{r} + a_{23} \\ a_{31} I + a_{32} \mathbf{r} + a_{33} \end{bmatrix},$$

(18)

where

$$a_{11} = I_1^T I_3^T, \quad a_{21} = I_1^T I_3^T, \quad a_{31} = I_1^T I_3^T,$$

$$a_{12} = \xi I_1^T c_1 I_1^T c_1 I_3^T + [I^T (c - c') l' T 0 0],$$

$$a_{22} = \xi I_1^T c_1 I_1^T c_1 I_3^T + [0 I^T (c - c') l' T 0 0],$$

$$a_{32} = \xi I_1^T c_1 I_1^T c_1 I_3^T + [0 0 I^T (c - c') l' T 0 0],$$

$$a_{13} = -\xi I_1^T e'' , \quad a_{23} = -\xi I_1^T e'' , \quad a_{33} = -\xi I_1^T e'' ,$$

(19)

and $I = [l_1, l_2, l_3]^T$, $e = e_1$, $e' = e_2$, and $e'' = e_1$ or $c_2$, depending on which trifocal set from the stereo correspondence across the two camera-matrix frames is considered. $I_{proj}$ is a function of the line measurements in the second and third view, and the form is derived by simply applying the trifocal tensor equations. The trifocal tensor incidence relation (4) then leads to

$$I_{proj} \times I = I^T [T_1, T_2, T_3] I' [I],$$

$$= \begin{bmatrix} a_{11} I + a_{12} \mathbf{r} + a_{13} \\ a_{21} I + a_{22} \mathbf{r} + a_{23} \\ a_{31} I + a_{32} \mathbf{r} + a_{33} \end{bmatrix} + \begin{bmatrix} a_{12} I - a_{22} \mathbf{r} \\ a_{21} I - a_{11} \mathbf{r} \\ a_{31} I - a_{11} \mathbf{r} \end{bmatrix} \mathbf{r} + \begin{bmatrix} a_{13} I - a_{33} \mathbf{r} \\ a_{23} I - a_{33} \mathbf{r} \\ a_{33} I - a_{33} \mathbf{r} \end{bmatrix} = A_{11} \mathbf{r} + A_{21} \mathbf{r} + A_{31} = 0.$$  

(20)

The equations are derived automatically using a symbolic computation toolbox and the code is submitted as supplementary material.

APPENDIX B

Here we outline the form of matrix $M^*$. With $q$, $s$ and $c$ provided, we substitute $A^*$ and $a^*$ into (14) from the paper, and obtain

$$M^* = \begin{pmatrix} a^* + a^*_0 - a^*_1 + a^*_2 \\ 2a^*_2 + 2a^*_3 \\ 2a^*_3 + 2a^*_4 \\ 2a^*_4 + 2a^*_5 \end{pmatrix}^T$$

(21)

using a symbolic computation toolbox. The code is submitted as supplementary material.

REFERENCES

[1] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in Proc. Int. Conf. Comput. Vis., Barcelona, Spain, Nov. 2011, pp. 2320–2327.

[2] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality, Nara, Japan, Nov. 2007, pp. 225–234.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” IEEE Trans. Robotic., vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[4] G. Hartley, “Projective reconstruction from line correspondences,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, 1994, pp. 903–907.

[5] C. J. Taylor and D. J. Kriegman, “Structure and motion from line segments in multiple images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 11, pp. 1021–1032, Jun. 1995.

[6] R. Hartley and A. Zisserman, Multiple View Geometry Computer Vision, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2004.

[7] A. Bartoli and P. Sturm, “Structure-from-motion using lines: Representation, triangulation, and bundle adjustment,” Comput. Vis. Image Understand., vol. 100, no. 3, pp. 416–441, Dec. 2005.

[8] A. Bartoli and P. Sturm, “Structure-from-motion using lines: Representation, triangulation, and bundle adjustment,” Comput. Vis. Image Understand., vol. 100, no. 3, pp. 416–441, Dec. 2005.

[9] G. Schindler, P. Krishnamurthy, and F. Dellaert, “Line-based structure from motion for urban environments,” in Proc. 3rd Int. Symp. 3D Data Process., Visualization, Transmiss. (3DPVT), Chapel Hill, CA, USA, Jun. 2006, pp. 846–853.

[10] E. Eade and T. Drummond, “Edge landmarks in monocular SLAM,” Image Vis. Comput., vol. 27, no. 5, pp. 588–596, Apr. 2009.

[11] P. Smith, I. Reid, and A. J. Davison, “Real-time monocular SLAM with straight lines,” in Proc. Brit. Mach. Vis. Conf., Edinburgh, U.K., 2006, pp. 17–26.
A. P. Gee and W. Mayol-Cuevas, “Real-time model-based SLAM using...

A. Elqursh and A. Elgammal, “Line-based relative pose estimation,” in...

O. Johannsen, A. Sulc, and B. Goldluecke, “On linear structure from motion for light field cameras,” in...

A. Handa, T. Whelan, J. McDonald, and A. J. Davison, “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM,” in...

X. Zuo, X. Xie, Y. Liu, and G. Huang, “Robust visual SLAM with point and line features,” 2017, arXiv:1711.08654, [Online]. Available: http://arxiv.org/abs/1711.08654

Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, “PL-VIO: Tightly-coupled monocular visual-inertial odometry using point and line features,” Sensors, vol. 18, no. 4, p. 1159, 2018.

J. Bian, X. Hui, X. Zhao, and M. Tan, “A Point-Line-Based SLAM framework for UAV close proximity transmission tower inspection,” in Proc. IEEE Int. Conf. Robot. Biomimetics (ROBio), Dec. 2018, pp. 1016–1021.

X. Luo, Z. Tan, and Y. Ding, “Accurate line reconstruction for point and line-based stereo visual odometry,” IEEE Access, vol. 7, pp. 185108–185120, 2019.

S.-J. Li, B. Ren, Y. Liu, M.-M. Cheng, D. Frost, and V. A. Prisacariu, “EDLines: A real-time line segment detector with a false detection control,” Pattern Recognit. Lett., vol. 32, no. 13, pp. 1633–1642, Oct. 2011.

L. Zhang and R. Koch, “An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency,” J. Vis. Commun. Image Represent., vol. 24, no. 7, pp. 794–805, Oct. 2013.

C. Akinlar and C. Topal, “Multi-resolution surfel maps for efficient dense 3D modeling and tracking,” J. Vis. Commun. Image Represent., vol. 25, no. 1, pp. 137–147, Jan. 2014.

J. Stäckler and S. Behnke, “Multi-resolution surfel maps for efficient dense 3D modeling and tracking,” J. Vis. Commun. Image Represent., vol. 24, no. 7, pp. 794–805, Oct. 2013.

J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Oct. 2016, pp. 501–518.
PEIHONG YU received the B.S. degree from Shanghai University, Shanghai, China, in 2016. She is currently pursuing the master’s degree with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and ShanghaiTech University, Shanghai. Her research interests include 3-D vision, SLAM, and robotics.

CEN WANG received the B.S. degree from the Dalian University of Technology, Shanghai, China, in 2017. She is currently pursuing the master’s degree with ShanghaiTech University, Shanghai. Her research interests are computer vision, computer graphics, and machine learning.

ZHIRUI WANG received the bachelor’s degree from the Beijing Institute of Technology, Beijing, China, and the bachelor’s and M.Phil. degrees from Australian National University, ACT, Australia, in 2016 and 2019, respectively. He is currently a Software Engineer at Motovis Intelligent Technologies Ltd. His research interests are focusing on structures from motion and SLAM.

JINGYI YU (Member, IEEE) received the B.S. degree from Caltech, in 2000, and the Ph.D. degree from MIT, in 2005. He is the Vice Dean of the School of Information Science and Technology, ShanghaiTech University. He is also affiliated with the University of Delaware. He has published over 120 articles at highly refereed conferences and journals, including over 70 articles at premier conferences and journals such as CVPR/ICCV/ECCV/TPAMI. He has also been granted ten U.S. patents. His research has been generously supported by the National Science Foundation (NSF), the National Institute of Health, the Army Research Office, and the Air Force Office of Scientific Research (AFOSR). His research interests span a range of topics in computer vision and computer graphics, especially on computational photography, non-conventional optics, and camera design. He is a recipient of the NSF CAREER Award, the AFOSR YIP Award, and the Outstanding Junior Faculty Award from the University of Delaware. He has served as the General Chair, the Program Chair, and the Area Chair of many international conferences such as CVPR, ICCV, ECCV, ICCP, and NIPS. He will be the Program Chair of ICPR and the IEEE CVPR 2021. He is currently an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, and Elsevier CVIU.

LAURENT KNEIP (Member, IEEE) received the Dipl.Ing. degree from Friedrich-Alexander University Erlangen–Nürnberg and the Ph.D. degree from ETH Zurich. He worked at the Autonomous Systems Lab, ETH Zurich. He was an Associate Investigator of the Australian Centre of Excellence for Robotic Vision. He currently holds a position as an Assistant Professor tenure-track with the School of Information Science and Technology, ShanghaiTech University, where he founded and directs the Mobile Perception Laboratory (MPL). He has published more than 60 articles mostly in top conferences and journals such as ICCV/CVPR/ECCV/ICRA/TPAMI/TRO. He is the main author of OpenGV, the open-source project. His long-term contributions in geometric computer vision have, for example, been awarded the Marr Prize (honorable mention) by ICCV, in 2017. He was a recipient of the Discovery Early Career Researcher Award (DECRA) from the Australian Research Council, hosted by Australian National University.

***