Camera Pose Estimation Using 2D-3D Line Pairs Acquired and Matched with a Robust Line Detector and Descriptor

PhD Thesis

Hichem Abdellali

Supervisor:
Prof. Zoltan Kato

Doctoral School of Computer Science
Institute of Informatics
University of Szeged

Szeged
2021

Abstract

Camera pose estimation refers to estimating the camera pose, which is composed of the rotation $\mathbf{R}$ and translation $\mathbf{t}$ parameters with respect to the world coordinate system. Estimating the projective mapping and thereby extracting the camera parameters is the goal of camera pose estimation. However, the pose estimation process requires input parameters, like points, planes, or lines. In this thesis, we work with 2D-3D line pairs; therefore, we focused on finding a solution for 2D line detection and matching through fully automatic algorithms and CNN.

This thesis proposes novel solutions for pose estimation using 2D-3D line pairs and a novel line segment detector and descriptor based on convolutional neural networks. The pose solvers can estimate the absolute and relative pose of a camera system of a general central projection camera such as perspective or omnidirectional cameras. They work both for the minimal case and the general case using 2D-3D line pairs in presence of noise, or outliers. The algorithms have been validated on a large synthetic dataset as well as on real data. Experimental results confirm the stable and real-time performance under realistic conditions. Comparative tests show that our method compares favorably to the latest State-of-the-Art algorithms. Regarding the learnable line segment detector and descriptor, it allows efficient extraction and matching of 2D lines on perspective images. While many hand-crafted and deep features have been proposed for key points, only a few methods exist for line segments. However, line segments are commonly found in structured environments, in particular urban scenes. Moreover, lines are more stable than points and robust to partial occlusions. Our method relies on a 2-stage deep convolutional neural network architecture: In stage 1, candidate 2D line segments are detected, and in stage 2, a descriptor is generated for the extracted lines. The network is trained in a self-supervised way using an automatically collected dataset. Experimental results confirm the State-of-the-Art performance of the proposed L2D2 network on two well-known datasets for autonomous driving both in terms of detected line matches as well as when used for line-based camera pose estimation and tracking.

Abstract In Hungarian

A kamera-pozíció becslése a kamera pózának becslésére vonatkozik, amely az $\mathbf{R}$ forgatási és a $\mathbf{t}$ transzlációs paraméterekből áll a világkoordináta-rendszerhez képest. A kamerapóz becslés célja a projektív leképezés becslése és ezáltal a kamera paramétereinek kinyerése. A pózbecslési folyamat azonban bemeneti adatokat igényel, melyek lehetnek például pontok, síkok vagy vonalak. Ebben a dolgozatban 2D-3D egyenespárokkal dolgozunk, ezeket használó megoldásokat javasoltunk, ezért megoldást kerestünk a 2D egyenes-detekcióra és -illesztésre is egy teljesen automatikus algoritmus és egy CNN segítségével.

A disszertáció új megoldásokat javasol a pózbecsléshez 2D-3D egyenespárok és egy új, konvolúciós neurális hálón alapuló egyenes-szakasz detektor és leíró segítségével. A pózbecslők meg tudják becsülni egy általános középpontos kamerarendszer abszolút és relatív pozícióját, ilyen kamerarendszer állhat például perspektivikus vagy omnidirekcionális kamerákból is. A megoldók 2D-3D vonalpárok használatával mind a minimális esetekre, mind az általános esetekre működnek, zaj vagy kiugró értékek jelenlétében is. Az algoritmusokat nagyméretű szintetikus adatkészleten és valós adatokon is validáltuk. A kísérleti eredmények alátámasztják a stabil és valós idejű teljesítményt akár valós körülmények között is. Az összehasonlító tesztek azt mutatják, hogy módszerünk jól teljesít a legmodernebb algoritmusokkal szemben. Ami a tanulható egyenes-szakasz detektort és leírót illeti, lehetővé teszi a 2D szakaszok hatékony kinyerését és illesztését perspektivikus képeken. Míg számos kézzel készített és mély-háló alapú jellemzőt javasoltak már kulcspontokhoz, a vonalszakaszokhoz csak néhány módszer létezik. A vonalszakaszok azonban gyakran megtalálhatók strukturált környezetekben, különösen városi jelenetekben. Ezenkívül a vonalak stabilabbak, mint a pontok, és robusztusak a részleges takarásokra. Ezért fontosak az olyan alkalmazásokhoz, mint a pózbecslés, a vizuális odometria vagy a 3D-s rekonstrukció. Módszerünk egy 2 lépéses mélykonvolúciós neurális hálózati architektúrán alapul: az 1. szakaszban lehetséges 2D vonalszakaszokat detektálunk, a 2. szakaszban pedig leírót generálunk a kinyert vonalakhoz. A hálózat tanítása önfelügyelt módon történik egy automatikusan gyűjtött adatkészlet segítségével. A kísérleti eredmények megerősítik a javasolt L2D2 hálózat legkorszerűbb teljesítményét két jól ismert autonóm vezetési adathalmazon, mind a detektált vonalak párosítása, mind pedig az egyenes alapú kamerapózbecslési és -követési alkalmazások tekintetében.

Introduction

Computer vision is understood as the host of techniques to acquire, process, analyze, and understand complex higher-dimensional data from our environment for scientific and technical exploration [15]. Simply computer vision aims to create a model of the real world using cameras for analysing or understanding. This discipline became a key technology in many areas, from industrial usage to simple end users. In this thesis we focus on the camera pose estimation topic. In the literature, the camera pose estimation is an essential step in many applications, including robotics, navigation and 3D reconstruction. The problem is also known as extrinsic camera calibration. It addresses the issue of determining the position and orientation of a camera with respect to a world coordinate frame. Mainly the pose estimation refers to two cases, absolute and relative camera poses. In this thesis, novel solutions for the absolute and the relative camera pose estimation problem are presented [Abdellali and Kato, (2018)][Abdellali et al., (2019)a][Abdellali et al., (2019)b]. The proposed technique estimates the camera pose using 2D-3D line pairs for central perspective and omnidirectional cameras. Moreover, a solution to acquire the 2D-3D lines that are, required by any real world applications is presented, also focusing on repeatable lines, that are good for pose estimation through a fully automatic process, relying on a convolutional neural network to detect and match 2D lines [Abdellali et al., (2021)].

Camera Pose Estimation with 2D-3D Line Pairs

In this section, the 3 proposed solutions for the pose estimation problem using 2D-3D line pairs are explained, the 3 solutions are based on the same geometrical observation, that uses the relation between 3D lines and their corresponding 2D lines observed by a camera. Each of the 3 solutions uses different solving derivation to create robust solvers. In [Abdellali and Kato, (2018)], we exploit the vertical direction that can be obtained from devices like an IMU sensor, to formulate a solver that works on perspective images, and solving the absolute and relative pose simultaneously. Nevertheless, prior information from IMU are not always available. Hence, in [Abdellali et al., (2019)a], we explored a novel way to formulate a solver for a perspective multi-view camera system, using an intermediate coordinate system built from the 2D-3D line pairs, this solver has been improved and tuned with a 3D data normalisation, a line back-projection and a refinement step. However, perspective cameras cover a smaller field of view compared to omnidirectional cameras. Thus, we formulate a solution in [Abdellali et al., (2019)b], that is composed of a direct minimal solver and a direct least squares solver that works on central cameras such as perspective and omnidirectional cameras, thanks to the spherical model of [30, 31]. Here, the 3D lines are represented as $L = (\mathbf{V}, \mathbf{X})$, where $\mathbf{V}$ is the unit direction vector of the line and $\mathbf{X}$ is a point on the line [12, 34]. The projection of L in a multi-view camera system produces one 2D line $l_i, i = 0, 1 \dots N$ in each image plane, which can also be represented as $l_i = (\mathbf{v}_i, \mathbf{x}_i), i = 0, 1 \dots N$. Intuitively, for each camera i (i = 0,1$\dots N$), both L and l_i lie on a projection plane π_i passing through the camera projection center $\mathbf{C}_i$ (see Fig. 1). The unit normal to the plane π_i in the camera coordinate system $\mathcal{C}_i$ is denoted by $\mathbf{n}_i$, which can be computed from the image line l_i as :

$$\mathbf{n}_i = \frac{(\mathbf{v}_i \times \mathbf{x}_i)}{\|\mathbf{v}_i \times \mathbf{x}_i\|} \tag{1}$$

For the reference camera $\mathcal{C}_0$, L lies also on π_i, and its direction vector $\mathbf{V}$ is perpendicular to $\mathbf{n}_i$. Hence, we get the following equation which involves only the absolute pose $(\mathbf{R}, \mathbf{t})$ [12]

$$\mathbf{n}_0^\top \mathbf{R}\mathbf{V} = \mathbf{n}_0^\top \mathbf{V}^{\mathcal{C}_0} = 0, \tag{2}$$

where $\mathbf{R}$ is the rotation matrix from the world coordinate frame $\mathcal{W}$ to the reference camera

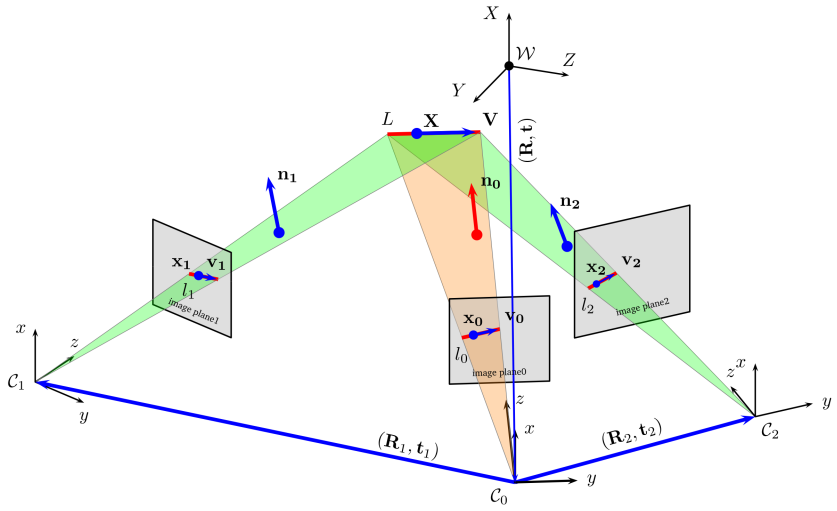

Figure 1. *Projection of a 3D line in a 3 cameras system [Abdellali and Kato, (2018)].*

$\mathcal{C}_0$ frame and $\mathbf{V}^{\mathcal{C}_0}$ denotes the unit direction vector of L in the reference camera coordinate frame. Furthermore, the vector from the camera center $\mathbf{C}_0$ to the point $\mathbf{X}$ on line L is also lying on π_0, thus it is also perpendicular to $\mathbf{n}_0$:

$$\mathbf{n}_0^\top(\mathbf{R}\mathbf{X}+\mathbf{t}) = \mathbf{n}_0^\top \mathbf{X}^{\mathcal{C}_0} = 0, \tag{3}$$

where $\mathbf{t}$ is the translation from the world coordinate frame $\mathcal{W}$ to the reference camera $\mathcal{C}_0$ frame and $\mathbf{X}^{\mathcal{C}_0}$ denotes the point $\mathbf{X}$ on L in the reference camera coordinate frame. In the case of N cameras, equations for the relative poses are derived similarly as equation (2) and (3) such as a 3D line L has up to N images. Note that for the other cameras we will have two equations for each other camera: one similar to (2) that contains the rotation from the reference camera and the rotation from the other camera; and another equation similar to (3) that includes the rotations and translations from both the reference and the other camera.

For central cameras [7, 18, 24, 29] and [Abdellali et al., (2019)b], a 3D line L is centrally projected by a projection plane $\pi_L = (\mathbf{n}, d)$ onto the surface of the unit sphere $\mathcal{S}$ (see Fig. 2). Since the camera projection center is also on π_L, d becomes zero and thus π_L is uniquely determined by its unit normal $\mathbf{n}$. The image of L is the intersection of the *ray surface* $\mathcal{S}$ and π_L, which is a *great circle*, while a particular line segment becomes a *great circle segment* on the unit sphere $\mathcal{S}$ with endpoints $\widetilde{\mathbf{a}}$ and $\widetilde{\mathbf{b}}$ (both are on $\mathcal{S}$, hence they are unit length!). The unit normal $\mathbf{n}$ to the projection plane π_L in the camera coordinate frame $\mathcal{C}$ is then given by: $\mathbf{n} = \widetilde{\mathbf{a}} \times \widetilde{\mathbf{b}}$. Until this point we mentioned the core equations which will let us build the proposed solvers.

Camera Pose Estimation with Known Vertical Direction

We formulate the first solution [Abdellali and Kato, (2018)] based on the previously presented geometric observations, and when the vertical direction is known. A lot of cheap available IMUs provide very accurate roll and pitch angle, i.e. the vertical direction. Here we show that knowing the camera vertical direction can reduce the complexity of the rotation matrix and simplify the camera pose problem [20]. Assuming that the camera coordinate system is a standard right-handed system with the X axis pointing up, the coordinates of the world vector $(1,0,0)^\top$ are known in the camera coordinate frame $\mathcal{C}$. Given this *up-vector*, we can compute the rotation $\mathbf{R}_v = \mathbf{R}_Z(\gamma)\mathbf{R}_Y(\beta)$ around Y and Z axes, which aligns the world X axis with the camera X axis, thus the only unknown parameter in the rotation

matrix $\mathbf{R} = \mathbf{R}_v\mathbf{R}_X(\alpha)$ is the rotation $\mathbf{R}_X(\alpha)$ around the vertical X axis. We aim to compute $\mathbf{R}_X(\alpha)$ and $\mathbf{t}$, *i.e.* 4 unknowns: the rotation angle α and the translation components t_x, t_y, t_z of each camera using equation (2) for the absolute pose and and similarly as (2) for relative poses. Although each 2D-3D line correspondence $L \leftrightarrow l$ provides 2 equations, only one contains $\mathbf{t}$. Therefore we need at least 3 line correspondences for each camera, *i.e.* in the minimal case, we need 6 2D–3D line pairs to solve for the absolute and relative pose of the stereo camera system.

In order to eliminate $\cos(\alpha)$ and $\sin(\alpha)$ from $\mathbf{R}_X(\alpha)$, let us substitute $q = \tan(\alpha/2)$ [6, 12, 20], for which $\cos(\alpha) = (1-q^2)/(1+q^2)$ and $\sin(\alpha) = 2q/(1+q^2)$, yielding to a new form for $\mathbf{R}_X$.

Minimal Case: Herein, our system of equation [Abdellali and Kato, (2018)] involves only the absolute pose, and only one relative pose. Substituting the new form of $\mathbf{R}$ (after we eliminate $\cos(\alpha)$ and $\sin(\alpha)$) into (2), we get a quadratic equation in terms of q:

$$\mathbf{n}_0^\top \mathbf{R}_v \mathbf{R}_X(q)\mathbf{V} = q^2 A_1 + qB_1 + C_1 = 0 \tag{4}$$

where A_1, B_1, C_1 are coefficients in terms of $\mathbf{n}_0$, $\mathbf{R}_v$, and $\mathbf{V}$. For the relative pose, we back-substitute both $\mathbf{R} = \mathbf{R}_v\mathbf{R}_X(q)$ as well as $\mathbf{R}_1 = \mathbf{R}_{v,1}\mathbf{R}_X(q_1)$ yielding

$$\begin{aligned} \mathbf{n}_1^\top \mathbf{R}_{v,1}\mathbf{R}_X(q_1)\mathbf{R}_v\mathbf{R}_X(q)\mathbf{V} = \\ q^2q_1^2A + q^2q_1B + q^2C + qq_1D + qE + qq_1^2F + q_1^2G + q_1H + I = 0 \end{aligned} \tag{5}$$

where the coefficients A through I are expressed in terms of $\mathbf{n}_1$, $\mathbf{R}_v$, $\mathbf{R}_{v,1}$, and $\mathbf{V}$. Equations (4) and (5) provide a system of quadratic equations for the unknown absolute rotation q and relative rotation q_1. This system can be efficiently solved using a direct solver generated by [19], which gives 2 solutions for q and 2 for q_1. For each possible $(\mathbf{R}, \mathbf{R}_1)$ pair, the absolute and relative translations $\mathbf{t}$ and $\mathbf{t}_1$ are obtained by backsubstituting the rotation matrices into (3) yielding a system of linear equations in terms of $\mathbf{t}$ and $\mathbf{t}_1$, which can be solved by SVD decomposition.

Multi-view Case: We formulate a least-squares solution for the multi-view case [Abdellali and Kato, (2018)] based on equation (4) and equation (5) by simply stacking the equations of each camera, including the absolute rotation q and the relative rotation q_i. A least-squares solution is obtained by finding a solution $(q, q_1, \ldots, q_N)$ which minimizes the squared error of the system. This would yield an 8^{th} order polynomial system of equation in terms of $(q, q_1, \ldots, q_N)$

$$\begin{aligned} (q^2A_1 + qB_1 + C_1)^2 = 0 \\ \forall i = 1, \ldots, N: \quad (q^2q_i^2A + q^2q_iB + q^2C + qq_iD + qE + qq_i^2F + q_i^2G + q_iH + I)^2 = 0 \end{aligned} \tag{6}$$

The first equation of (6) contains only q, hence its derivative is

$$A_3q^3 + B_3q^2 + C_3q + D_3 = 0, \tag{7}$$

while the second equation of (6) contains both q and q_i yielding 2 equations, which are the partial derivatives with respect to q. For a camera system with N cameras, we obtain a total of $2(N-1)+1$ equations. This provides us with one initial relative pose for each camera and $N-1$ possible value for the absolute pose, which are averaged to have one initial value for the absolute pose. Starting from this initialization, the system of the polynomial equations is efficiently solved in MATLAB via `fsolve`.

Camera Pose Estimation for A Perspective Camera System

Here [Abdellali et al., (2019)a], we deal with the case when IMU sensors are not accessible, and we use the full rotation matrix. In order to reduce the number of unknowns to 2 in (2), we eliminate one rotation in $\mathbf{R}$ by defining an intermediate coordinate system $\mathbb{N}$ [38, 41, 43][Abdellali et al., (2019)a] in which the rotation angle around the X axis can be easily obtained. Let us select a line pair (L_0, l_0) with the longest projection length. The origin of $\mathbb{N}$ is located at the origin of $\mathcal{W}$ and its axes $(\mathbf{X}_\mathcal{M}, \mathbf{Y}_\mathcal{M}, \mathbf{Z}_\mathcal{M})$ are

$$\mathbf{Y}_\mathcal{M} = \frac{\mathbf{n}_0^\mathcal{C}}{\|\mathbf{n}_0^\mathcal{C}\|} \tag{8}$$

$$\mathbf{X}_\mathcal{M} = \frac{\mathbf{n}_0^\mathcal{C} \times \mathbf{V}_0^\mathcal{W}}{\|\mathbf{n}_0^\mathcal{C} \times \mathbf{V}_0^\mathcal{W}\|} \tag{9}$$

$$\mathbf{Z}_\mathcal{M} = \frac{\mathbf{X}_\mathcal{M} \times \mathbf{Y}_\mathcal{M}}{\|\mathbf{X}_\mathcal{M} \times \mathbf{Y}_\mathcal{M}\|} \tag{10}$$

where the Y axis of $\mathbb{N}$ aligns with $\mathbf{n}_0^\mathcal{C}$. The rotation $\mathbf{R}_\mathcal{M} = [\mathbf{X}_\mathcal{M}, \mathbf{Y}_\mathcal{M}, \mathbf{Z}_\mathcal{M}]^\top$ rotates the normals and direction vectors into the intermediate frame $\mathbb{N}$. The rotation $\mathbf{R}_x^\mathcal{M}$ around X axis within $\mathbb{N}$ is then easily calculated because it is the angle between the Z axis and $\mathbf{V}_0^\mathcal{M}$, hence the rotation matrix acting within the intermediate coordinate frame $\mathbb{N}$ is composed of the rotations around the remaining two axes as

$$(1+s^2)(1+r^2)\mathbf{R}^\mathcal{M} = \mathbf{R}_y^\mathcal{M}(s)\mathbf{R}_z^\mathcal{M}(r) = \begin{bmatrix} (1-s^2)(1-r^2) & -2r(1-s^2) & 2s(r^2+1) \\ 2r(s^2+1) & (s^2+1)(1-r^2) & 0 \\ -2s(1-r^2) & 4sr & (1-s^2)(r^2+1) \end{bmatrix} \tag{11}$$

and we obtain the new form of (2) as: $(\mathbf{R}_\mathcal{M}\mathbf{n})^\top \mathbf{R}^\mathcal{M}(\mathbf{R}_x^\mathcal{M}\mathbf{R}_\mathcal{M}\mathbf{V}) = \mathbf{n}^{\mathcal{M}\top}\mathbf{R}^\mathcal{M}\mathbf{V}^\mathcal{M} = 0$.

Expanding the previous equation gives a 4-th order polynomial of (s, r) with coefficients in terms of $\mathbf{V}^\mathbb{N}$ and $\mathbf{n}^\mathbb{N}$: $\mathbf{a}^\top\mathbf{u} = 0$, where $\mathbf{V}^\mathbb{N}$ and $\mathbf{n}^\mathbb{N}$ are the 3D line unit direction vector and the projection plane unit normal, respectively, defined in the intermediate coordinate system $\mathbb{N}$. Each line pair generates one such equation, yielding a system of n equations, which is solved in the least squares sense. Thus the solution of the system of 2 polynomial equations provides the rotation parameters (s, r) [Abdellali et al., (2019)a]. Herein, we use automatic generator of Kukelova [19] to generate a solver using Grobner basis[19, 22]. Once the solution(s) are obtained, the complete $\mathbf{R}$, acting between the world coordinate frame $\mathcal{W}$ and the camera $\mathcal{C}$ frame, is obtained as $\mathbf{R} = \mathbf{R}_\mathcal{M}^\top(\mathbf{R}^\mathcal{M}\mathbf{R}_x^\mathcal{M})\mathbf{R}_\mathbb{N}$. The translation $\mathbf{t}$ is then obtained by backsubstituting $\mathbf{R}$ into (3) yielding a system of linear equations, which can be solved by SVD decomposition. We might have several solutions, the solver will only return the geometrically valid [1, 12, 23, 38, 41]. For the multi-view case, the projection of the 3D line L yields similar equations but the unknown relative pose $(\mathbf{R}_i, \mathbf{t}_i)$ will also be involved:

$$(\mathbf{R}_{\mathcal{M}_i}\mathbf{n}_i)^\top \mathbf{R}^{\mathcal{M}_i}(\mathbf{R}_x^{\mathcal{M}_i}\mathbf{R}_{\mathcal{M}_i}\mathbf{R}\mathbf{V}) = \mathbf{n}^{\mathcal{M}_i\top}\mathbf{R}^{\mathcal{M}_i}\mathbf{V}^{\mathcal{M}_i} = 0 \tag{12}$$

which –after a similar derivation as in the single camera case– yields also a system of polynomial equations, hence the same solver can be used to solve for each camera $\mathcal{C}_i, i = 1, \ldots, N-1$. Once the solutions are obtained, each $\mathbf{R}_i$ is backsubstituted into the corresponding linear system similar to (3) which is solved for $\mathbf{t}_i$ by SVD. To filter outlier line pairs we used the proposed solver within RANSAC [8], and we used the error measure proposed in [23].

For the pose refinement, we will now formulate a least-squares refinement for the multi-view case, based on the equations (2) by simply stacking the equations for each line pair in $\mathcal{C}_0$ and for each camera $i = 1, \ldots, N-1$ and each line pair in $\mathcal{C}_i$ containing the absolute pose $(\mathbf{R}, \mathbf{t})$ and the relative poses $(\mathbf{R}_i, \mathbf{t}_i)$. A least-squares solution is then obtained by minimizing the squared error of the system, which can be solved via standard algorithms like *Levenberg-*

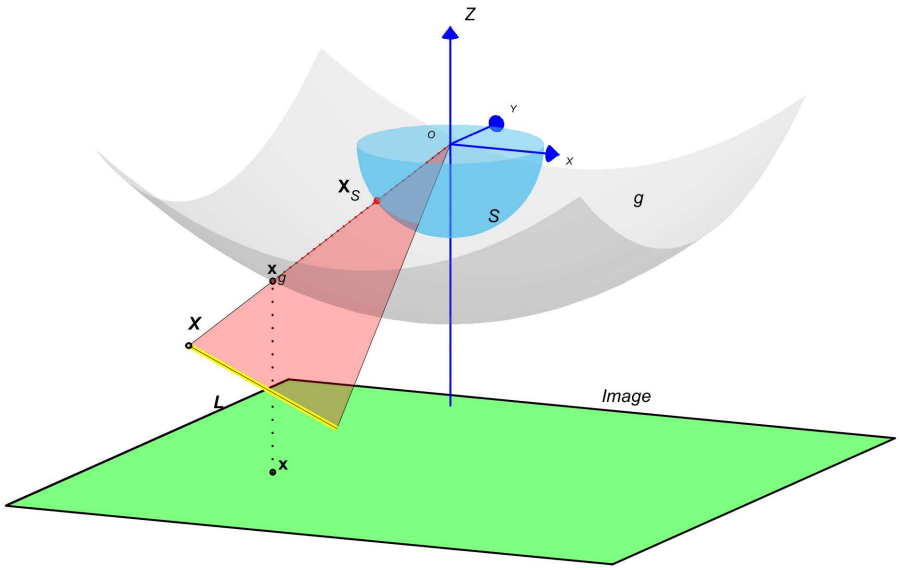

Figure 2. *Projection plane of a line (yellow) in the spherical camera model [Abdellali et al., (2019)b].*

Marquardt with the initialization obtained from the direct solver. Note, that this step is optional, and only executed for the overdetermined $n > 3$ case if the line parameters are noisy.

Pose Estimation using General Central Projection Cameras

We propose here a universal solution for central camera setups [Abdellali et al., (2019)a]. The solution is composed of a minimal direct solver using Grobner basis which works with 3 line pairs. Then a direct least squares solver which works for $n \geq 3$ 2D-3D line pairs. Both solvers run efficiently due to the low-order polynomial system of equations obtained via Cayley parametrization of the rotation matrix. Here, we adopted the Scaramuzza model that describe the distortion on the omnidirectional images, using parameters that describe the image projection function by a polynomial based on Taylor series expansion. Following Scaramuzza model [30, 31], we assume that the camera coordinate system is in the unit sphere $\mathcal{S}$ (see Fig. 2), the origin is the effective projection center of the omnidirectional camera. The omnidirectional camera projection is fully described by means of unit vectors $\mathbf{X}_\mathcal{S}$ in the half space of $\mathbb{R}^3$ and these points correspond to the unit vectors of the projection rays. Similarly, the image points of a perspective camera can be represented on the unit sphere $\mathcal{S}$ by the bijective mapping $\mathbf{x} \mapsto \mathbf{x}_\mathcal{S}$: $\mathbf{x}_K = \mathbf{K}^{-1}\mathbf{x}$ and $\mathbf{x}_\mathcal{S} = \mathbf{x}_K / \|\mathbf{x}_K\|$. Thus the projection of a calibrated central camera is fully described by means of unit vectors $\mathbf{x}_\mathcal{S}$ in the half space of $\mathbb{R}^3$. A 3D world point $\mathbf{X}$ is projected into $\mathbf{x}_\mathcal{S} \in \mathcal{S}$ by a simple central projection taking into account the pose:

$$\mathbf{x}_\mathcal{S} = \frac{\mathbf{R}\mathbf{X} + \mathbf{t}}{\|\mathbf{R}\mathbf{X} + \mathbf{t}\|} \tag{13}$$

We used the Cayley transform to obtain a parametrization of the rotation matrix $\mathbf{R}$ in terms of 3 parameters $\mathbf{b} = [b_1, b_2, b_3]^\top$. Following [11, 40], the Cayley transform of a rotation matrix is a skew-symmetric matrix and vice versa. Therefore the correspondence $\mathbf{R} \leftrightarrow [\mathbf{b}]_\times$ is a one-to-one map between skew-symmetric matrices (represented as 3-vectors) and 3D rotations, excluding rotation angles $\pm 180°$. Thus we have:

$$(1 + \mathbf{b}^\top\mathbf{b})\mathbf{R} = (1 - \mathbf{b}^\top\mathbf{b})\mathbf{I} + 2[\mathbf{b}]_\times + 2\mathbf{b}\mathbf{b}^\top = \begin{bmatrix} 1 + b_1^2 - b_2^2 - b_3^2 & 2b_1b_2 - 2b_3 & 2b_1b_3 + 2b_2 \\ 2b_1b_2 + 2b_3 & 1 - b_1^2 + b_2^2 - b_3^2 & 2b_2b_3 - 2b_1 \\ 2b_1b_3 - 2b_2 & 2b_2b_3 + 2b_1 & 1 - b_1^2 - b_2^2 + b_3^2 \end{bmatrix} \tag{14}$$

Minimal Solver: Using the Cayley parametrization of the rotation matrix $\mathbf{R}$, we get a second order polynomial equation from (2) where $\mathbf{n} = [n_1, n_2, n_3]^\top$ and $\mathbf{V} = [v_1, v_2, v_3]^\top$ are the projection plane unit normal and the 3D line unit direction vector, respectively. Given 3 such line-pairs, we obtain a system of 3 equations with the 3 unknown rotation parameters $\mathbf{b} = [b_1, b_2, b_3]^\top$, which can be easily solved by a solver using Grobner basis [17, 19, 21]. We used the automatic generator of Kukelova [19] for MATLAB and Kneip's generator [17] which produces a solver in C++, that is an order of magnitude faster! The translation $\mathbf{t}$ is then obtained by backsubstituting $\mathbf{R}$ into (3) yielding a system of linear equations, which can be solved by SVD decomposition. The solver returns valid poses $(\mathbf{R}, \mathbf{t})$. For RANSAC, we used a line back-projection error which works on the the unit sphere $\mathcal{S}$. Note that the 2D image data is normalized by definition as we work on the unit sphere, and the 3D lines are normalized for numerical stability [10].

Direct Least Squares Solver: For the direct least squares solver, we start from equation (2) and (3), each line pair generates one such pair of equations, yielding a system of $n > 3$ equations, which is solved in the least squares sence. Taking the sum of squares of the nonlinear system constructed with the basic equations [Abdellali et al., (2019)a] and then find $\arg\min_{\mathbf{b}} E(\mathbf{b})$. The first order optimality condition is

$$\nabla E(\mathbf{b}) = \begin{bmatrix} \frac{\partial E(\mathbf{b})}{\partial b_1} \\ \frac{\partial E(\mathbf{b})}{\partial b_2} \\ \frac{\partial E(\mathbf{b})}{\partial b_3} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \mathbf{d_{b_1}}_i^\top \mathbf{x_{b_1}} \\ \sum_{i=1}^{n} \mathbf{d_{b_2}}_i^\top \mathbf{x_{b_2}} \\ \sum_{i=1}^{n} \mathbf{d_{b_3}}_i^\top \mathbf{x_{b_3}} \end{bmatrix} = \mathbf{0} \tag{15}$$

where for each line pair $\mathbf{d_{b_1}}$, $\mathbf{d_{b_2}}$, and $\mathbf{d_{b_3}}$ can be expressed in terms of coefficients of each line pair. Thus the solution of the system of 3 polynomial equations (each of them is third order) in (15) provides the rotation parameters $\mathbf{b}$.
We successfully used solver generators of [17, 19] to generate a MATLAB and C++ solver for the above polynomial system. The translation $\mathbf{t}$ is then obtained by backsubstituting $\mathbf{R}$ into (3) yielding a system of linear equations, which can be solved by SVD decomposition. Multiple solutions are eliminated in the same way as for the minimal solver. Relative camera poses $(\mathbf{R}_i, \mathbf{t}_i)$ are also obtained in a similar way once the absolute pose is computed.

Overview and Results

Three novel solutions for the camera pose estimation using 2D-3D line pairs are proposed. Experimental tests on large synthetic as well as real datasets confirm the State-of-the-Art performance of the proposed solutions. Comparative results show that our method outperforms recent alternative methods (AlgLS [26], SRPnL [38]) in terms of speed, accuracy, and robustness. Furthermore, unlike these methods, our solutions work for multi-view scenarios and are robust to outliers when combined in a RANSAC-like method. As an example of a real data application: the Cayley solvers have been used, with perspective and omnidirectional images in a fusion application on Fig. 3.

Learnable Line Detector and Descriptor

In this section, a new *Learnable Line Detector and Descriptor* (L2D2) is developed [Abdellali et al., (2021)], the solution is robust enough to detect and match 2D lines across wide view-point changes. Our deep convolutional net's architecture consists of two stages:

1. A line segment detector with a lightweight residual network architecture inspired by wireframe [13]

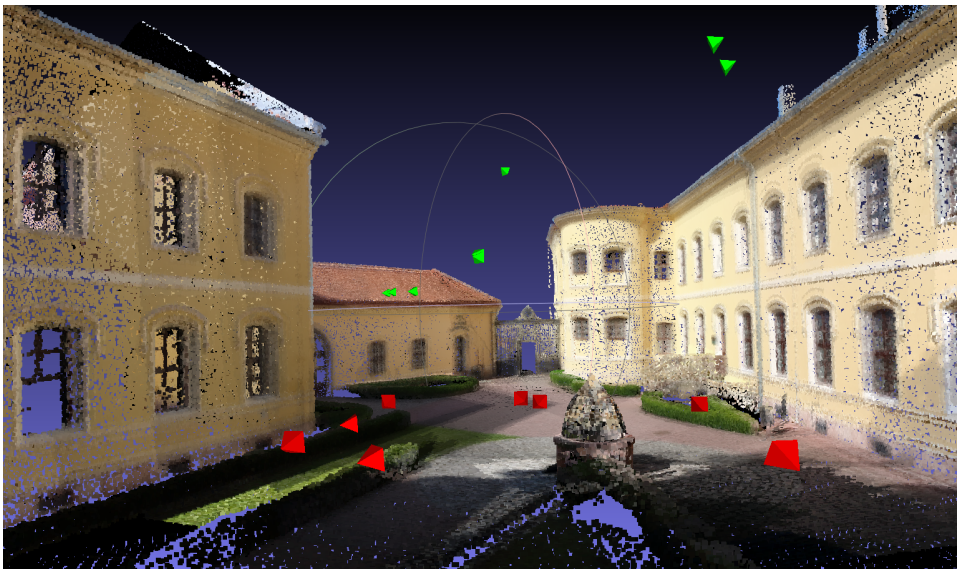

Figure 3. *Fusion result with the Cayley solvers shown as colorized pointcloud with estimated omni (red) and perspective (green) camera positions illustrated [Abdellali et al., (2019)b].*

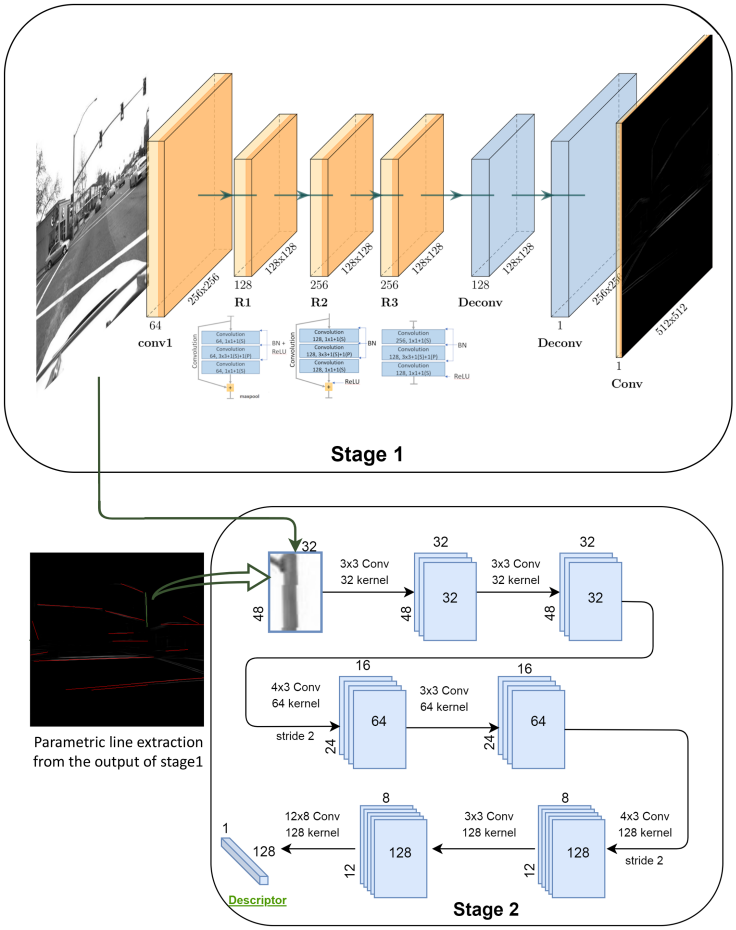

Figure 4. *Architecture of the proposed L2D2 network.* **Stage1***: Detector,* **Stage 2***: Descriptor. [Abdellali et al., (2021)]*

2. A patch-based descriptor network inspired by RAL-Net [42] with a rectangular patch size adapted to line-based orientation normalization, yielding 128-dimensional unit feature vectors that can be matched via an angular distance.

repeatable line is an infinite line, parts of which can be detected on multiple images. This is an important difference w.r.t keypoints or wireframes, where exact position of the point feature or line segment is critical. This higher level aspect is rarely considered in current line detectors. We argue that detecting less but more relevant line segments is better, as this way the descriptor could also be more reliable than matching too many irrelevant lines. An important part of our method is the automatic construction of a large training dataset of matching 2D line segment pairs. For this purpose, a dataset with ground truth camera poses and corresponding 3D point cloud is needed. Herein, the Lyft dataset [16] has been used. To create 2D line pairs, first 2D-3D line correspondences were determined by detecting lines on the images and projecting them into the 3D point cloud. Using a fully automatic algorithm outlined in the thesis [Abdellali et al., (2021)], all the relevant information can be extracted from the data structure, such as what are the 2D views of each 3D line (*i.e.* which camera sees the 3D line), or which 3D lines are visible on a given 2D image. These 2D line pairs are then used for the detector training, while the line support regions (patches) are extracted for all 2D lines corresponding to the same 3D line, yielding a list of all possible patch-correspondences of the 3D line used for the descriptor training. This list is then used to create the batches for training.

Statistics/Detector	L2D2	SOLD2	EDLines
detected line segments	**73,063**	70,836	66,887
unique infinite detected lines	**59,395**	53,381	44,485
percentage	**81.29%**	75.36%	66.51%
validated line segments	10,685	13,552	**17,771**
unique infinite validated lines	9,785	11,771	**13,762**
percentage	**91.58%**	86.86%	77.44%

Table 1. *Detector performance comparison [Abdellali et al., (2021)].*

Line Detector Network: The network architecture of [Abdellali et al., (2021)] is inspired by the line detection branch of the Wireframe Parsing [13] method. The architecture's complexity can be greatly reduced from 20.77M to only 1M parameters, while the line detection performance is similar. For training the line detector network, a mean squared difference loss is applied on each output of the batches of $b = 20$ images:

$$\mathcal{L}_{MSE} = \frac{1}{b}\sum_{i=1}^{b}(\frac{1}{N}\sum_{j=1}^{N}(h_j(I_i) - GT_j(I_i))^2) \tag{16}$$

where $h_j(I_i)$ is the j^{th} pixel of the detection heatmap output of the network for image I_i and $GT_j(I_i)$ is the corresponding binary image pixel of the ground truth lines on the image, while N is the number of pixels on the image. Each batch is constructed by starting from a randomly selected 3D line, collecting the camera views that see that line, then adding new lines from the field of view of the selected cameras, and repeating the process until the desired batch size is reached. We used the Stochastic Gradient Descent (SGD) optimizer for 100 epochs, and we set a fixed learning rate of 0.025, and the momentum to be 0.9, weight decay equals to 0.0001.

Line Descriptor Network: Given a set of 2D lines $\{l_i\}_{i=1}^{\ell}$, our approach consists of a support region selection mechanism which guarantees a normalized orientation with respect to each line l_i and a deep neural network architecture based on RAL-Net [42] (which adopts the HardNet [27] architecture, which is identical to L2Net [35]). Fig. 4 summarizes the

Descriptor/Detector	L2D2	SOLD2	EDLines
L2D2	**84.08% (5398/6420)**	**72.11% (5843/8103)**	**84.21% (9760/11590)**
SOLD2	82.49% (5296/6420)	70.15% (5685/8103)	78.04% (9045/11590)
SMSLD	72.35% (4645/6420)	65.36% (5296/8103)	74.78% (8667/11590)
DLD	77.71% (4989/6420)	66.74% (5408/8103)	76.20% (8832/11590)

Table 2. *Descriptor performance comparison on validated line segments [Abdellali et al., (2021)].*

layers. The input is the 32×48 pixels line support region with normalized grayscale values (subtracting the mean and dividing by the standard deviation) and the output is an L2 normalized 128D unit length descriptor. The whole feature extraction is built of full convolution layers, downsampling by two-stride convolution. There is a Batch Normalization (BN) [14] layer and a ReLu [28] activation layer in every layer except the last one. To prevent overfitting, there is a 0.3 Dropout layer [32] above the bottom layer as in RAL-Net [42]. We follow the HardNet [27] strategy to construct batches: First a matching set $M = \{l_{\mathbf{a}_i}, l^+_{\mathbf{a}_i}\}_{i=1}^N$ of N line pairs is generated, where $l_{\mathbf{a}_i}$ stands for an anchor line and $l^+_{\mathbf{a}_i}$ for its positive pair (*i.e.* they correspond the the same 3D line). M must contain exactly one pair originating from a given 3D line! Then the line support regions are extracted and passed through our L2D2 network. That provides the descriptors $(\mathbf{a}_i, \mathbf{p}_i)$, from which a pairwise $N \times N$ distance matrix $\mathbf{D}$ is calculated such that $\mathbf{D}_{i,j} = d(\mathbf{a}_i, \mathbf{p}_j), i = 1..N, j = 1..N$. Following [37, 42], we will use cosine similarity for our metric learning, since our descriptors are unit vectors:

$$d(\mathbf{a}_i, \mathbf{p}_j) = (1 - \mathbf{a}_i \cdot \mathbf{p}_j) = (1 - \cos(\sphericalangle(\mathbf{a}_i, \mathbf{p}_j))) \tag{17}$$

Using $\mathbf{D}$, for each matching pair $\mathbf{a}_i$ and $\mathbf{p}_i$, the closest non-matching descriptor $\mathbf{n}_i$ is found by searching the minimum over the off-diagonal elements of the ith row and ith column of $\mathbf{D}$. The following loss is then minimized for each batch:

$$\frac{1}{N}\sum_{i=1}^{N}(1 + \tanh(d(\mathbf{a}_i, \mathbf{p}_i) - d(\mathbf{a}_i, \mathbf{n}_i))) \tag{18}$$

The training data is partitioned into 3332 batches of 128 corresponding patch pairs, each batch being created from a cluster. We applied the strategy which trains for 200 epochs with learning rate linearly decreasing to 0 in the end, as in RAL-Net [42]. We choose Stochastic Gradient Descent (SGD) as our optimizer and we set the initial learning rate to be 0.1, and the momentum to be 0.9, dampening equal to 0.9 and weight decay equal to 0.0001.

Overview and Results

The efficiency and performance of the proposed Detector/Descriptor/Full Detector-Descriptor have been confirmed on different datasets such as Lyft [16], KITTI[9] and KITTI360[25] and on different testing contexts. The proposed solution outperforms the State-of-the-Art method (see Table 1, Table 2) in terms of matching detected repeatable lines. The proposed Detector-Descriptor [Abdellali et al., (2021)] was also combined with the robust solver of [Abdellali et al., (2019)b] for camera pose estimation and camera pose tracking applications which outperforms over the State-of-the-Art methods.

Summary of the Author's Contributions

Computer vision is understood as the host of techniques to acquire, process, analyze, and understand complex higher-dimensional data from our environment for scientific and technical exploration [15]. Simply computer vision aims to create a model of the real world using cameras for analyzing and understanding it. This discipline became a key technology in many areas, from industrial usage to simple end customers. In recent years, perceptual interfaces [36] have emerged to motivate an increasingly large amount of research within the machine vision community; some of the areas are structure from motion, stereo matching, text recognition, person tracking algorithms, camera calibration, stereo vision, point cloud segmentation, and pose estimation of rigid, articulated, and flexible objects. [33, 39]. Nowadays, the topic of pose estimation in real-time performance is in the center of interest for both the academic and industrial side, especially for autonomous driving in urban environments where pose estimation is needed to navigate in such complex space, it is essential to allow moving devices like a car, drone or a robot with one or multiple cameras to navigate and avoid obstacles. The pose estimation topic is fundamental in various computer vision applications, such as simultaneous localisation, and mapping (SLAM), image-based localization and navigation, augmented reality. This work presents my research on developing solutions for the problem of pose estimation using 2D-3D line pairs and also for the detection/acquisition and matching of such lines in a semi-supervised manner through CNN.

A.1 Key Points of the Thesis

In the following, I summarized my results and highlighted key findings in two main thesis groups. In the first one, I present my findings on the topic of pose estimation using 2D-3D line pairs known as the PnL problem, while in the second one, I present my results on the 2D line detection and 2D line matching topic. In Table A.1, the connections between the thesis points and the corresponding publications are displayed.

I Pose Estimation Using 2D-3D Line Pairs

The basic idea here, is to build up a system of polynomial equations whose solution directly provides the pose. This idea can be extended into a general framework for the pose estimation of a central spherical camera system composed of perspective and omnidirectional cameras. The following points summarize my contribution on the absolute pose

estimation topic:

(a) I have proposed a new solution inspiring from [12], on which we don't assume a known relative pose. In the proposed solution [Abdellali and Kato, (2018)], I derived the concrete equations, I have constructed the minimal solver and the least squares formulation of the equations for the general case. I have generated the synthetic noisy data, and performed the evaluation both on synthetic and real data. In the second solution called MRPnL [Abdellali et al., (2019)a]. I derived the concrete equations for both minimal and the general case. I experimentally tested, validated and plotted the performance of the two proposed solutions through quantitative and qualitative evaluation on synthetic and real data, respectively, and compared it with the Stat-of-the-Art methods.

(b) Given a Cayley representation of the rotation, I derived the concrete equations that involve the absolute and relative poses [Abdellali et al., (2019)b]. The solution works on a system of perspective and omnidirectional cameras. I have constructed the direct solvers using the automatic generator of [19] and [17]. I ran and demonstrated the performance of the solution through quantitative evaluations and proved its robustness against noise on the synthetic data and real data. I have also compared the results to the State-of-the-Art methods.

II 2D Line Detection and Matching

This thesis summarizes my contributions with a new line detector and descriptor [Abdellali et al., (2021)], that is a crucial ingredient of any pose estimation application. First, inspiring from the architecture of the line detection branch of the wireframe parsing [13, 44], I have proposed a lightweight line detection architecture. Second, I have adapted [42] to take a centered rectangular patch from the rotated detected line into a vertical orientation. I have implemented a fully automatic algorithm that can be applied to work with any new dataset to create high-quality training. I trained the full network using training data built by the proposed data generator. I measured the efficiency of the proposed network for line detection and matching performance. I demonstrated the performance of the proposed method against the State-of-the-Art methods.

	I		II
	a	b	
[Abdellali and Kato, (2018)]	•		
[Abdellali et al., (2019)a]	•		
[Abdellali et al., (2019)b]	•	•	
[Abdellali et al., (2021)]			•

Table A.1. *The connection between the thesis points and publications.*

Publications

Refereed Conference Papers

[Abdellali and Kato, (2018)] Hichem Abdellali and Zoltan Kato. Absolute and Relative Pose Estimation of a Multi-View Camera System using 2D-3D Line Pairs and Vertical Direction. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, Canberra, Australia, December 2018. IEEE. doi: doi:10.1109/DICTA.2018.8615792

[Abdellali et al., (2019)a] Hichem Abdellali, Robert Frohlich, and Zoltan Kato. A Direct Least-Squares Solution to Multi-View Absolute and Relative Pose from 2D-3D Perspective Line Pairs. In *Proceedings of ICCV Workshop on 3D Reconstruction in the Wild*, pages 2119–2128, Seoul, Korea, October 2019. IEEE. doi: doi:10.1109/ICCVW.2019.00267

[Abdellali et al., (2019)b] Hichem Abdellali, Robert Frohlich, and Zoltan Kato. Robust Absolute and Relative Pose Estimation of a Central Camera System from 2D-3D Line Correspondences. In *Proceedings of ICCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, pages 895–904, Seoul, Korea, October 2019. IEEE. doi: doi:10.1109/ICCVW.2019.00118

[Abdellali et al., (2021)] Hichem Abdellali, Robert Frohlich, Viktor Vilagos, and Zoltan Kato. L2D2: Learnable Line Detector and Descriptor. In *Proceedings of International Conference on 3D Vision*. IEEE, November 2021, Accepted.

Journal publications

[Abdellali and Kato, (2020)] Hichem Abdellali and Zoltan Kato. 3D reconstruction with depth prior using graph-cut. *Central European Journal of Operations Research*, 29(2):387–402, jul 2020. doi: 10.1007/s10100-020-00694-6

Bibliography

[1] Hichem Abdellali and Zoltan Kato. Absolute and Relative Pose Estimation of a Multi-View Camera System using 2D-3D Line Pairs and Vertical Direction. In *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, Canberra, Australia, December 2018. IEEE. doi: doi:10.1109/DICTA.2018.8615792.

[2] Hichem Abdellali and Zoltan Kato. 3D reconstruction with depth prior using graph-cut. *Central European Journal of Operations Research*, 29(2):387–402, jul 2020. doi: 10.1007/s10100-020-00694-6.

[3] Hichem Abdellali, Robert Frohlich, and Zoltan Kato. A Direct Least-Squares Solution to Multi-View Absolute and Relative Pose from 2D-3D Perspective Line Pairs. In *Proceedings of ICCV Workshop on 3D Reconstruction in the Wild*, pages 2119–2128, Seoul, Korea, October 2019. IEEE. doi: doi:10.1109/ICCVW.2019.00267.

[4] Hichem Abdellali, Robert Frohlich, and Zoltan Kato. Robust Absolute and Relative Pose Estimation of a Central Camera System from 2D-3D Line Correspondences. In *Proceedings of ICCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, pages 895–904, Seoul, Korea, October 2019. IEEE. doi: doi: 10.1109/ICCVW.2019.00118.

[5] Hichem Abdellali, Robert Frohlich, Viktor Vilagos, and Zoltan Kato. L2D2: Learnable Line Detector and Descriptor. In *Proceedings of International Conference on 3D Vision*. IEEE, November 2021.

[6] Cenek Albl, Zuzana Kukelova, and Tomás Pajdla. Rolling shutter absolute pose problem with known vertical direction. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3355–3363, Las Vegas, NV, USA, June 2016. doi: 10.1109/CVPR.2016.365. URL `http://dx.doi.org/10.1109/CVPR.2016.365`.

[7] Federico Camposeco, Torsten Sattler, and Marc Pollefeys. Minimal solvers for generalized pose and scale estimation from two rays and one point. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proceedings of European Conference Computer Vision*, volume 9909 of *Lecture Notes in Computer Science*, pages 202–218, Amsterdam, The Netherlands, October 2016. Springer. doi: 10.1007/978-3-319-46454-1_13. URL `http://dx.doi.org/10.1007/978-3-319-46454-1_13`.

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. doi: 10.1145/358669.358692. URL `http://doi.acm.org/10.1145/358669.358692`.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2012. doi: 10.1109/cvpr.2012.6248074.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2004.

[11] Richard Hartley, Jochen Trumpf, Yuchao Dai, and Hongdong Li. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, July 2013.

[12] Nora Horanyi and Zoltan Kato. Multiview absolute pose using 3D - 2D perspective line correspondences and vertical direction. In *Proceedings of ICCV Workshop on Multiview Relationships in 3D Data*, pages 1–9, Venice, Italy, October 2017. IEEE.

[13] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 626–635, 2018. doi: 10.1109/CVPR.2018.00072.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of International Conference on Machine Learning*, page 448–456. JMLR.org, 2015.

[15] Bernd Jähne. *Handbook of computer vision and applications*. Academic Press, San Diego, 1999. ISBN 9780123797728.

[16] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. urlhttps://level5.lyft.com/dataset/, 2019.

[17] Laurent Kneip. *Polyjam*, 2015 [online]. https://github.com/laurentkneip/polyjam.

[18] Laurent Kneip, Hongdong Li, and Yongduek Seo. UPnP: an optimal O(n) solution to the absolute pose problem with universal applicability. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of European Conference Computer Vision, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 127–142, Zurich, Switzerland, September 2014. Springer.

[19] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *Proceedings of European Conference on Computer Vision*, pages 302–315. Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-88690-7_23.

[20] Zuzana Kukelova, Martin Bujnak, and Tomáš Pajdla. Closed-form solutions to minimal absolute pose problems with known vertical direction. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Proceedings of Asian Conference on Computer Vision, Part II*, volume 6493 of *LNCS*, pages 216–229, Queenstown, New Zealand, November 2010. Springer.

[21] Viktor Larsson, Kalle Astrom, and Magnus Oskarsson. Efficient solvers for minimal problems by syzygy-based reduction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2383–2392. IEEE, jul 2017. doi: 10.1109/cvpr.2017.256.

[22] Viktor Larsson, Magnus Oskarsson, Kalle Åström, Alge Wallis, Zuzana Kukelova, and Tomás Pajdla. Beyond grobner bases: Basis selection for minimal solvers. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 3945–3954, Salt Lake City, UT, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00415. URL `http://openaccess.thecvf.com/content_cvpr_2018/html/Larsson_Beyond_Grobner_Bases_CVPR_2018_paper.html`.

[23] Gim Hee Lee. A minimal solution for non-perspective pose estimation from line correspondences. In *Proceedings of European Conference on Computer Vision*, pages 170–185, Amsterdam, The Netherlands, October 2016. Springer.

[24] Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Motion estimation for self-driving cars with a generalized camera. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 2746–2753, Portland, OR, USA, June 2013. doi: 10.1109/CVPR.2013.354. URL `http://dx.doi.org/10.1109/CVPR.2013.354`.

[25] Yiyi Liao, Jun Xie, and Andreas Geiger. KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *arXiv.org*, 2109.13410, 2021.

[26] Faraz M. Mirzaei and Stergios I. Roumeliotis. Globally optimal pose estimation from line correspondences. In *2011 IEEE International Conference on Robotics and Automation*, pages 5581–5588. IEEE, may 2011. doi: 10.1109/icra.2011.5980272.

[27] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, pages 4826–4837. Curran Associates, Inc., 2017.

[28] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of International Conference on Machine Learning*, page 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[29] Robert Pless. Using many cameras as one. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–587, 2003.

[30] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701. IEEE, oct 2006. doi: 10.1109/iros.2006.282372.

[31] Davide Scaramuzza, Agostino Martinelli, and Roland Siegwart. A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In *International Conference on Computer Vision Systems*, pages 45–51, Washington, USA, January 2006.

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[33] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer London, 2011. ISBN 978-1-84882-935-0. doi: 10.1007/978-1-84882-935-0.

[34] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17 (11):1021–1032, November 1995. ISSN 0162-8828. doi: 10.1109/34.473228. URL `http://dx.doi.org/10.1109/34.473228`.

[35] Yurun Tian, Bin Fan, and Fuchao Wu. L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6128–6136. IEEE, jul 2017. doi: 10.1109/cvpr.2017.649.

[36] Matthew Turk. Computer vision in the interface. *Communications of the ACM*, 47(1): 60–67, jan 2004. doi: 10.1145/962081.962107.

[37] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 5265–5274. IEEE, June 2018.

[38] Ping Wang, Guili Xu, Yuehua Cheng, and Qida Yu. A novel algebraic solution to the perspective-three-line problem. *Machine Vision and Applications*, 2019. in press.

[39] Christian Wöhler. *3D Computer Vision: Efficient Methods and Applications*. Springer London, 2 edition, 2013. ISBN 978-1-4471-4150-1. doi: 10.1007/978-1-4471-4150-1.

[40] F. C. Wu, Z. H. Wang, and Z. Y. Hu. Cayley transformation and numerical stability of calibration equation. *International Journal of Computer Vision*, 82(2):156–184, dec 2008. doi: 10.1007/s11263-008-0193-x.

[41] C. Xu, L. Zhang, L. Cheng, and R. Koch. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1209–1222, 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2582162.

[42] Yanwu Xu, Mingming Gong, Tongliang Liu, Kayhan Batmanghelich, and Chaohui Wang. Robust angular local descriptor learning. In C.V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler, editors, *Proceedings of Asian Conference on Computer Vision*, pages 420–435, Perth, Australia, 2019. Springer. ISBN 978-3-030-20873-8.

[43] Xiaohu Zhang, Zheng Zhang, You Li, Xianwei Zhu, Qifeng Yu, and Jianliang Ou. Robust camera pose estimation from unknown or known line correspondences. *Applied Optics*, 51(7):936, feb 2012. doi: 10.1364/ao.51.000936.

[44] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In *Proceedings of International Conference on Computer Vision*, pages 962–971. IEEE, October 2019.