Fachhochschule
Münster University of
Applied Sciences

Report

"Motion Estimation for Self-Driving Cars With a Generalized Camera"

Authoress / Author

Yves-Noel Weweler <y.weweler@fh-muenster.de>

March 2, 2016, Steinfurt

Department:	Electrical Engineering and Computer Science
Degree:	Master of Computer Science
Advisor:	Prof. Dr.-Ing. Jürgen te Vrugt

Contents

List of Figures

Abstract

This report will give a brief review of the paper "Motion Estimation for Self-Driving Cars With a Generalized Camera" [1]. The task of estimating the position and orientation of a mobile system in space is a often found problem in modern machinery that has numerous solutions. In the past odometry as it is often called, has especially evolved with the raise of GPS and affordable small electronics like gyroscopes. Today such technologies are not only used in the fields of robotics and aeronautics but also for cars and even for mobile phones. In Respect of autonomous vehicles there is the need to have additional sources of information to determine the position of a car and enable compensation of the drawbacks of the other techniques. Therefore I have decided to take a closer look at a visual based solution as described in [1]. Egomotion or visual odometry as it is often called, is a convenient approach based on images made by cameras that enable the extraction of such information. Based on knowledge of the field of stereo vision one can extract the orientation and route a object took in space from sequential images.

1 Introduction

Today there exist plenty of techniques for estimating the motion and orientation of mobile systems because it is a often found problem not only for cars but for all kinds of modern machinery. In the past odometry as it is often called has especially evolved with the raise of GPS and affordable small electronics like gyroscopes. Today such technologies are not only used in the fields of top line commercial robotics and aeronautics but also for cars, mobile phones and even child toys. Visual systems are gaining more and more relevance at that, since they proved to be a rather inexpensive data source compared to expensive hardware like radar or lidar systems. Nowadays cameras are already found in cars for assistance systems like parking and digital driving monitors. But they are still less integrated for more complex tasks, because of several problems with processing their data. Today several algorithms still only work offline because of their high needs for processing power or specially crafted hardware solutions. Low frame rates and complex algorithms designed to only work with specific cameras, prevent a broad deployment. Other problems with existing algorithms is their robustness against dynamically changing scenes or huge occlusions.

With "Motion Estimation for Self-Driving Cars With a Generalized Camera" [1] the task of developing a new egomotion algorithm capable of meeting real-time requirements for on-road cars was presented. Egomotion is a purely visual approach based on images made by cameras that enable the extraction of an objects motion in space. They employed the concept of an generalized-camera system to reach independence from the camera setups construction. To reduce efforts involved in estimating a cars motion, they described the cars motion using the Ackermann steering principle and hence constrained it to only circular motion in a plane. They derive a two-point minimal solution to solve for the relative motion between to camera frames. In addition, investigations on the effects of degeneracy are provided and a solution is developed to deal with them. They provide a real-world dataset investigation as a proof of concept for the capabilities of the developed approach.

2 Fundamentals

2.1 Plücker Lines

When describing projections, a convenient way of representing directed 3D lines is needed. As explained in [2], there are several solutions for that. A finite length line for example could be described by specifying two endpoints of a line or if it has infinite length, by specifying any two points on the line. Another way would be to specify a random point on a line together with a direction vector. All this methods would require a six element vector to describe a 3D line. The problem however with this way of describing lines is, that there are infinite representations of a single line. Two identical lines my be specified differently, that would make in inconvenient to check whenever two lines are identical or what the exact distance between them is.

A better way would be to use the direction vector of the line and a vector describing the nearest point on the line from a origin. A related way is to start with an arbitrary point on the line and take its cross product with the direction vector. This cross product is independent of the point chosen and uniquely defines the line. The direction unit vector v along with this cross product of a point P on the line are known as Plücker coordinates and are denoted by a 6-vector

$$(v \quad P \times v)$$

Figure 1 illustrates a Plücker line described from a point and a direction vector.
Note that there are other variations and notations of using Plücker coordinates to describe lines. One alternative representations for finite lines is the endpoint model as given in [3]. In this model lines are described in a distinct way using its two endpoints in Plücker coordinate form.

2.2 Generalized Camera Model

The concept of an generalized camera will be described here, since its understanding is important in order to be able to follow the developed solution. Generalized camera systems allow using multiple cameras as if they were a single imaging device, even when they do not share a common center of projection. As stated in [4], this model abstracts away from exactly what path light takes as it passes through the lenses and mirrors of an arbitrary imaging system. Instead, it identifies each pixel with the region of space that affects that sensor. One reasonable model often used for that is cone estimation. A complete definition of this generalized model for a imaging systems was defined in terms of ray pixels (raxels) in [5]. They described a image taken by a generalized camera as a set of raxel measurements captured by the system. Figure 4 illustrates a generalized camera setup on a car. The setup is made

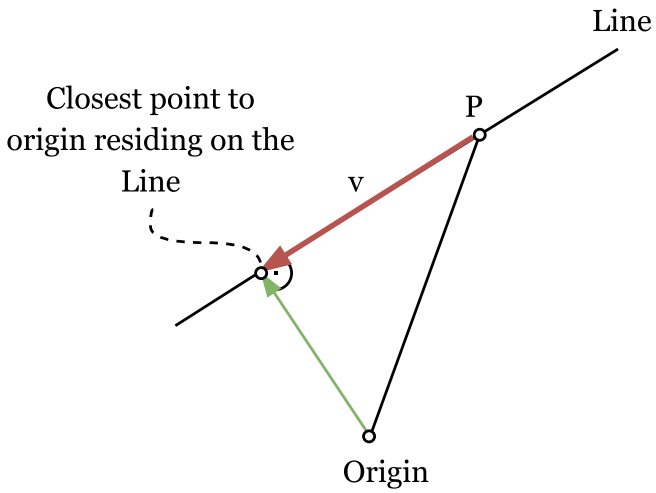


Figure 1: Example of a Plücker line created from a unit direction vector v and arbitrary point P on the line.

up out of individual cameras $C_1, C_2, \cdots, C_i$ at arbitrary locations on the car, where i numerates the cameras. This cameras are describes relative to a reference frame denoted by V. To describe this setup geometrically we need a way to represent the cameras parameters and its effects on the images. As it is the case with the common pinhole camera model, all cameras can be described by a set of transformations. To describe such a transformation and hence a camera, we need two different parameter sets. The intrinsic parameters of a camera describe the camera specific internal configuration. Common parameters are the focal length f_x, f_y, the skew coefficient γ and the cameras principal point u_0, v_0. In the following the intrinsic parameters will be denoted as

$$K_i = \begin{pmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Note that non-linear intrinsic parameters like the lens-distortion are important, but are not included in this linear model. If needed, they have to be dealt with separately. The extrinsic camera parameters denote a coordinate system transformation from the 3D world coordinates to the 3D camera coordinates. Therefore they describe the translation of the camera center and its rotation in the world. In what follows the extrinsic parameters will be denoted with the cameras rotation R_{C_i} and its translation t_{C_i} using a 4×4 matrix as followed

$$T_{C_i} = [R_{C_i}\ t_{C_i};\ 0\ 1]$$

The normalized image coordinate of a point x_{ij} for one camera is then given by $\hat{x}_{ij} = K_i^{-1} x_{ij}$, hence reverting the effects of the internal camera parameters.
6-vector Plücker lines are used to describe a light ray that connects an image coordinate x_{ij} and a 3D world point X_j. As described in subsection 2.1 a Plücker line denoted by l_{ij} is constructed of a direction 3-vector, here denoted by $u_{ij} = R_{C_i} \hat{x}_{ij}$ relative to the reference frame V and a point $(t_{C_i} \times u_{ij})^T$ the line passed through.

$$l_{ij} = \begin{bmatrix} u_{ij}^T & (t_{C_i} \times u_{ij})^T \end{bmatrix}^T$$

This allows description of X_j decoupled from a single center of projection.
A visual impression of the geometric construction of a camera, can be seen in Figure 2. Reformulation of the epipolar constraint for pinhole and projective cameras as shown in [4], allows formulation of a generalized essential 6×6 matrix E_{GC} and a generalized epipolar constraint shown in Equation 1.

$$l_{ij,k+1}^T \underbrace{\begin{bmatrix} E & R \\ R & 0 \end{bmatrix}}_{E_{GC}} l_{ij,k} = 0 \tag{1}$$

Where $l_{ij,k}$ and $l_{ij,k+1}$ are the corresponding Plücker lines from frame k and $k+1$ and E is the essential matrix from the conventional epipolar constraint as described in [6, p. 257]. Therefore $E = R\,[t]_x$ where t is the translation of the camera between two frames.

Note that t is only determined up to scale when using one camera, but the metric scale can be recovered using the generalized camera setup as described in subsection 3.4.

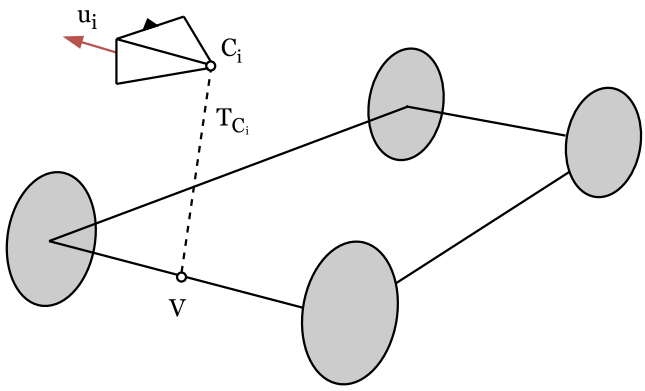


Figure 2: Construction of the camera system attached to a car. The camera C_i is related to the reference frame V by its translation t_{C_i} and its rotation R_{C_i}.

3 Motion Estimation

3.1 Approach

In this section a conceptual overview of the developed approach is given. Figure 3 shows the tasks involved to estimate the relative motion and orientation between to sets of images. First a set of synchronized images is shot with a generalized camera setup mounted on a car. Intra-camera correspondences are then computed from the images, matching the contents of each image together as described in subsection 3.2. Afterwards a solution for the position and yaw angle is computed from the correspondences as particularized in subsection 3.3 and fed into a 2-Point RANSAC algorithm. In the case the calculated yaw angle θ is found to be near zero, because of the degeneracy described in subsection 3.4, inter-camera correspondences are extracted between the images to compute the scale ρ using an 1-Point exhaustive search. This data is then further refined using non-linear filtering and Kalman filtering. Combining the calculated relative steps the full trajectory of the car can be approximated.

3.2 Point Correspondences

The algorithm described above, distinguishes between intra-camera and inter-camera point correspondences. Intra-camera point correspondences refer to points which are seen by the same camera over two consecutive frames. Inter-camera point correspondences on the other hand refer to frames which are seen by different cameras over two consecutive frames. Figure 4 illustrates intra and inter-camera correspondences.

Because of the circumstance that two consecutive frames recorded by the same camera often show the same part of a scene with just minor differences, one can find a lot of corresponding points between the frames. With two different cameras the amount of receivable correspondences is much lower. Therefore intra-camera point correspondences are used for calculation and in the case of a degeneracy when the car is moving straight, one-additional inter-camera correspondence is used to retrieve the scale. If there should be no additional inter-camera correspondence, the scale is propagated from the previous estimates using a Kalman filter.

Point correspondences are extracted and matched using Speeded Up Robust Features (SURF) [7] on the GPU instead of the Scale-invariant feature transform (SIFT) technique. Both SIFT and SURF are scale invariant feature detectors. Apparently SIFT appears to be the more accurate feature detector, but it underlies patent-restrictions and tends to have some drawbacks. SIFT is patented by David G. Lowe and can't be used for commercial products freely [8]. As reported in [9], SIFT sometimes has problems with the density and distribution of features in images. In scenes

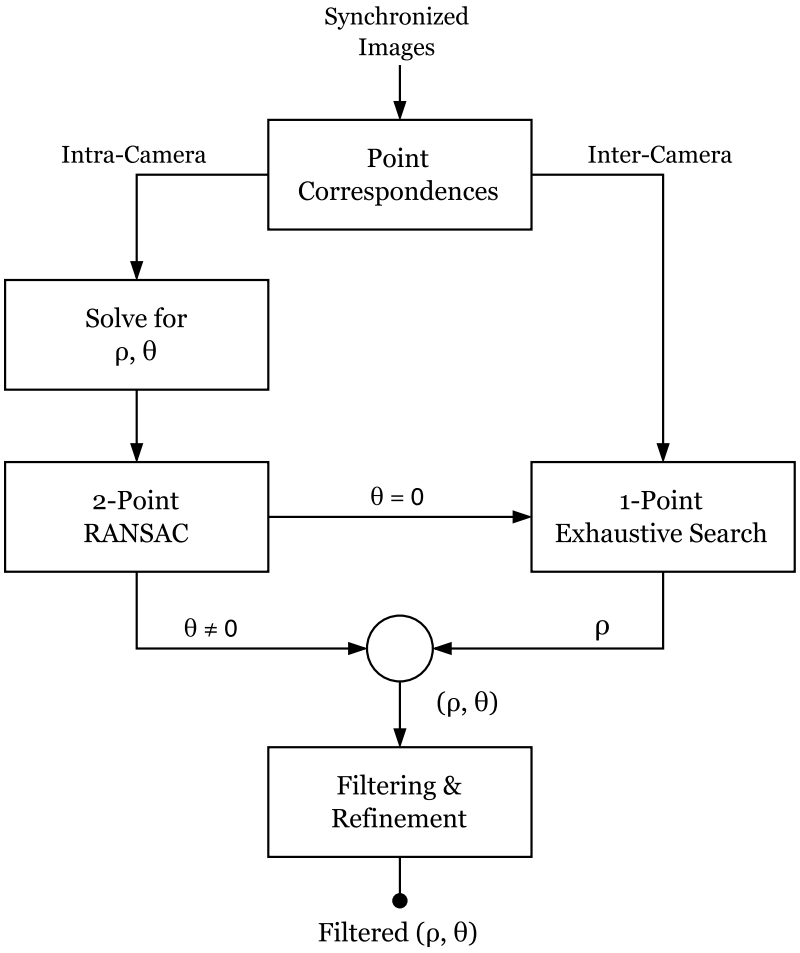


Figure 3: Overview of the developed approach for motion estimation using a generalized camera on a car.

where not a lot of features can be found, like it is the case with the balustrades of bridges or when only other cars and lamp posts can be seen, SIFT tends to deliver a lot less features. SURF act similar, but as the name suggest it offers speed in tradeoff of precision to enable realtime calculation, as it is needed for a usable egomotion approach. Other feasible methods to determine features are the Harris Corner Detection [10] or the Kanade-Lucas-Tomasi Feature Tracker (KLT) [11] which is often found in stereo applications.

Note, if needed, using a feature detection technique to determine point correspondences only allows for a sparse density point cloud reconstruction of the scene. If a feature rich and dense reconstruction would be needed, other more computational expensive matching techniques should be used in this step.

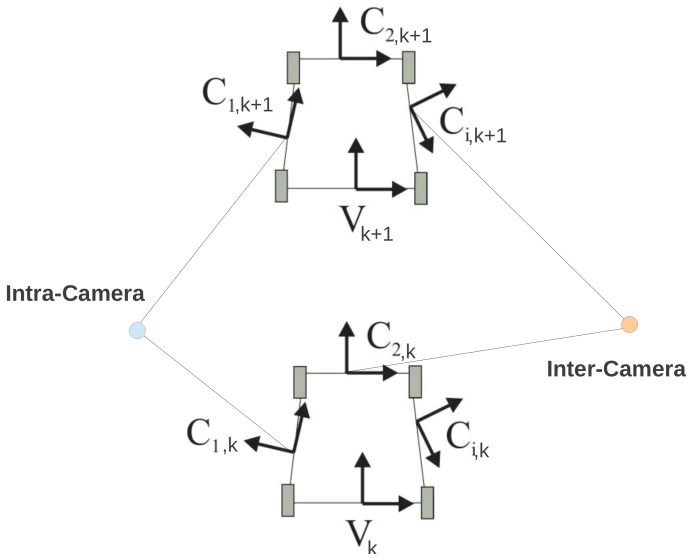


Figure 4: Example of intra- and inter-camera point correspondences. (Source: [1])

3.3 Point Minimal Solution

To be able to meet the realtime requirement for motion estimation, the motions freedom has to be constrained to a needed minimum. This allows for more accurate and faster outliner removal as given in [12] and is the prime factor for speeding up the trajectory recovery. Therefore we stick to the minimal suitable on-road model called the Ackermann steering principle, and constrain it to two parameters, to only allow circular motion in a plane [13]. In this model the front wheels are used for steering as it can be seen in Figure 5. The wheels are applied with different steering angles to allow for a smooth circular motion. For all cars following the Ackermann steering principle, there exists a ICR (Instantaneous Center of Rotation) the car moves around while steering. Therefore the cars motion ca be described by an angle and the radius from the ICR. A straight motion in this model can be represented along a circle with a infinite radius.

To determine the trajectory of a car, the relative motion between two consecutive frames V_k and V_{k+1} has to be recovered. Following the derivation for general planar motion from for Ackermann steering in [9], the relative rotation R and the relative translation t describe the relative circular motion in a plane. As it can be seen in Equation 2, because of the angle $\varphi = \frac{\theta}{2}$ illustrated in Figure 5, this motion is dependent on two basic parameters - scale ρ and yaw angle θ. These can be calculated using two Plücker line correspondence pairs as explained in subsubsection 3.3.2 and subsubsection 3.3.3.

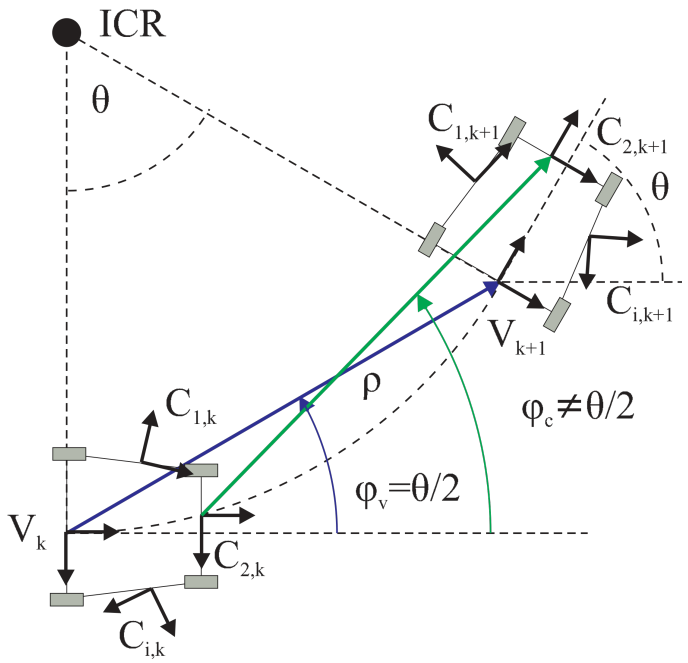


Figure 5: Ackermann steering principle for planar and circular motion of a car. (Source: [1])

$$R = \begin{bmatrix} cos\ \theta & -sin\ \theta & 0 \\ sin\ \theta & cos\ \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad t = \rho \begin{bmatrix} cos\ \varphi_v \\ sin\ \varphi_v \\ 0 \end{bmatrix} \tag{2}$$

There exist several other approaches to describe the motion a car can perform. A less constrained motion model, allows for a wider range of possible motions the car can perform, but increases the number of point correspondences and hence computational power needed.

3.3.1 Rewrite Generalized Epipolar Constraint

Using the standard essential matrix for pinhole cameras from [6, p. 257] and the rotation and translation from Equation 2 the generalized essential matrix E_{GC} can be written as shown in Equation 3. Combining these equations, allows do derive the epipolar constraint in terms of ρ and θ and therefore constrain the possible motion to the Ackermann steering.

$$E_{GC} = \begin{bmatrix} E & R \\ R & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \rho\ sin\ \frac{\theta}{2} & cos\ \theta & -sin\ \theta & 0 \\ 0 & 0 & -\rho\ cos\ \frac{\theta}{2} & sin\ \theta & cos\ \theta & 0 \\ \rho\ sin\ \frac{\theta}{2} & \rho\ cos\ \frac{\theta}{2} & 0 & 0 & 0 & 1 \\ cos\ \theta & -sin\ \theta & 0 & 0 & 0 & 0 \\ sin\ \theta & cos\ \theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

As the indices ij on the Plücker lines are not strictly needed for writing down the constraint, they are dropped for readability and the Plücker line vectors are denoted by $l = [u^T \ (t_C \times u)^T]^T$ for frame k and $l' = [u'^T \ (t_{C'} \times u')^T]^T$ for frame $k+1$. Expanding the generalized epipolar constraint from Equation 1, it can be written as

$$a\ cos\ \theta + b\ sin\ \theta + c\rho\ cos\ \frac{\theta}{2} + d\rho\ \frac{\theta}{2} + e = 0 \tag{4}$$

where the coefficients are

$$\begin{aligned} a &= -u_w(t_{Cx}u'_y - t_{Cy}u'_x) - u'_w(t_{C'x}u_y - t_{C'y}u_x) \\ &\quad + u_y(t_{C'w}u'_x - t_{Cw}u'_x) + u_x(t_{Cw}u'_y - t_{C'w}u'_y) \\ b &= u_x(t_{C'x}u'_w - t_{C'w}u'_x) + u_y(t_{C'y}u'_w - t_{C'w}u'_y \\ &\quad - u_x(t_{Cx}u_w - t_{Cw}u_x) - u'_y(t_{Cy}u_w - t_{Cw}u_y) \\ c &= u_w u'_y - u_y u'_w \\ d &= u_x u'_w + u_w u'_x \\ e &= u_w(t_{C'x}u'_y - t_{C'y}u'_x) + u'_w(t_{Cx}u_y - t_{Cy}u_x) \end{aligned} \tag{5}$$

The subscripts x, y and w refer to the components in the vectors. For more detailed calculation steps, see Appendix A.

Equation 4 forms the new generalized epipolar constraint with the Ackermann steering principle due to the use of the constrained rotation and translation.
As this new equations has two unknowns, two coefficient vectors $(a_1, b_1, c_1, d_1, e_1)$ and $(a_2, b_2, c_2, d_2, e_2)$ are needed to solve for the unknowns.

3.3.2 Solve for scale ρ

$$cos\ \theta = 1 - 2sin^2\frac{\theta}{2} \tag{6a}$$
$$sin\ \theta = 2sin\frac{\theta}{2}cos\frac{\theta}{2} \tag{6b}$$

Using the trigonomic half-angle formulas from Equation 6 and substitute $\alpha = cos\ \frac{\theta}{2}$ and $\beta = sin\ \frac{\theta}{2}$, the new GEC from Equation 4 can be transposed to solve for ρ as shown in Equation 7.

$$\rho = \frac{-e_1 - a_1(1 - 2\beta^2) - b_1(2\alpha\beta)}{c_1\alpha + d_1\beta} \tag{7a}$$
$$\rho = \frac{-e_2 - a_2(1 - 2\beta^2) - b_2(2\alpha\beta)}{c_2\alpha + d_2\beta} \tag{7b}$$

3.3.3 Solve for yaw angle θ

Equalizing the two equations for ρ from Equation 7 to eliminate ρ, we get

$$\begin{aligned}(2a_1\beta^2 - 2b_1\alpha\beta - e_1 - a_1)(c_2\alpha + d_2\beta)\\ -(2a_2\beta^2 - 2b_2\alpha\beta - e_2 - a_2)(c_1\alpha + d_1\beta) = 0\end{aligned} \tag{8}$$

while the Pythagorean identity from Equation 9 has to be satisfied.

$$sin^2\ \frac{\theta}{2} + cos^2\ \frac{\theta}{2} = \alpha^2 + \beta^2 = 1 \tag{9}$$

Using the Sylvester Resultant method we can now determine if the polynomials from Equation 8 and Equation 9 have the common non constant factor $\alpha = cos\ \frac{\theta}{2}$ we could eliminate to get a solvable equation in terms of $\beta = sin\ \frac{\theta}{2}$ [14]. The resultant of the two polynomials is a six degrees polynomial equation equal to the determinant of the sylvester matrix. This equation in terms of $\beta = sin\ \frac{\theta}{2}$, can be further reduced to a cubic polynomial by setting $\gamma = \beta^2$ as shown in Equation 10.

$$A\beta^6 + B\beta^4 + C\beta^2 + D = 0 \tag{10a}$$
$$A\gamma^3 + B\gamma^2 + C\gamma + D = 0 \tag{10b}$$

A, B, C and D are known coefficients made up of $(a_1, b_1, c_1, d_1, e_1)$ and $(a_2, b_2, c_2, d_2, e_2)$. The full expressions can be looked up in Appendix B. As described in the paper, there is an interesting behavior of D when using purely inter-camera correspondences ($t_C = t_{C'}$). As it can easily be seen in Equation 5 when putting $t_C = t_{C'}$, the last to terms of the coefficient a cancel out and $a = -e$. Putting this new relation in Equation 11, we see that all terms cancel out and $D = 0$.

$$\begin{aligned} D = &-c_2^2(e_1^2 + a_1^2) - 2c_2^2 e_1 a_1 - c_1^2(e_2^2 + a_2^2) - 2c_1^2 e_2 a_2 \\ &+ 2c_2 c_1(a_1 a_2 + e_1 e_2) + 2c_2 c_1(e_1 a_2 + a_1 e_2) \end{aligned} \tag{11}$$

Hence , Equation 10b becomes

$$\gamma(A\gamma^2 + B\gamma + C) = 0 \tag{12}$$

Using the cubic formula from Equation 13 the roots can be computed. Therefore the solutions for γ are 0 (as it can be seen from Equation 12) and the remaining two solutions can be computed using the cubic formula.

$$\gamma = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \tag{13}$$

Putting $\gamma = \beta^2$ back in to the relation, we get up to a maximum of six real solutions for β where two are always 0, $(+0, -0)$. Since $\beta = sin\ \frac{\theta}{2}$, the possible yaw angles are $\theta = 2\arcsin(\beta)$. The trajectory now can be recovered using ρ and θ, but should be filtered for outliners as described in subsection 3.5 to make the results more robust. As mentioned earlier one can get many more and especially more reliable correspondences with intra- than inter-camera correspondences. As shown above, there is also a mathematical benefit from doing so, because it is more efficient to compute the roots of the quadratic polynomial from Equation 12 than from a cubic polynomial from Equation 10b.

3.4 Degenerated Case: Metric Scale Computation

Since a fully calibrated camera rig for which the cameras intrinsics and extrinsics are known was used in the paper, the relative motion can be calculated with metric scale. But because intra-camera correspondences are used which are captured by a single camera, there is a degeneracy when the car is moving straight ($\theta = 0$). This can be observed by substituting $\theta = 0$ into Equation 7 where the numerator cancels out since $a = -e$ as it is the case for intra-camera

$$\begin{aligned}\rho &= \frac{-e_1 - a_1(1-2\beta^2) - b_1(2\alpha\beta)}{c_1\alpha + d_1\beta} \\ &= \frac{-e_1 + e_1(1-2sin^2\ 0) - 2b_1(cos\ 0\ sin\ 0)}{c_1 cos\ 0 + d_1 sin\ 0} \\ &= \frac{0}{c_1} = 0\end{aligned}$$

As proposed in the paper, one can still assign unit scale $\rho = 1$ for the solution of $\theta = 0$, because an unit scale still fulfills the Sampson error computation [15].

The scale can always be uniquely determined from the GEC when there is at least one inter-camera point correspondence. In this case one has $t_C \neq t_{C'}$ and $a \neq -e$ for the straight case $\theta = 0$, which shows when put into Equation 7

$$\begin{aligned}\rho &= \frac{-e_1 - a_1(1-2\beta^2) - b_1(2\alpha\beta)}{c_1\alpha + d_1\beta} \\ &= \frac{-e_1 - a_1(1-2sin^2\ 0) - 2b_1(cos\ 0\ sin\ 0)}{c_1 cos\ 0 + d_1 sin\ 0} \\ &= \frac{-e_1 - a_1}{c_1}\end{aligned}$$

that the scale can be recovered. Therefore one additional inter-camera correspondence can be used when $\theta = 0, \rho = 1$ turns out to be the solution with the highest inliners for the general intra-camera case after outliner removal, described in subsection 3.5. In practice, this can be done effectively by searching through all inter-camera point correspondences for inliners and then using them to calculate ρ.

3.5 Robust Estimation

As conducted in the paper the 2-point algorithm for estimating the scale and the yaw angle is made robust by implementing it within Random Sample Consensus (RANSAC) [12], to effectively reject outliners. RANSAC is used for interpreting and smoothing data that contains a significant percentage of gross errors. Fitting of models in the presence of many data outliners is ideally suited for applications in automated image analysis where interpretation is based on data provided by error-prone inputs like feature detectors. Classical techniques for parameter estimation, such as least squares, optimize the fit of a model to all of the presented data. They have no internal mechanisms for detecting and rejecting gross errors like RANSAC does [12].

Since the correspondences are not ideal, there are observation errors that have to be dealt with. Every line-correspondence l, l' that includes a observation error, breaks the epipolar constraint from Equation 1. Only a sub-pixel perfect correspondence could satisfy the constraint. The distance between the observed correspondence and the perfect one is called the reprojection error. The Sampson error is the approximated reprojection error with a linear approximation of the epipolar constraint. [15]

The Sampson error is used for evaluation of a fit using intra-camera correspondences in the RANSAC model. For a computational less expensive evaluation the normal essential matrix instead of the generalized one suits, because only intra-camera correspondences are used and it has to fulfill the constraint from Equation 14 where $\hat{x}'$ and $\hat{x}$ are normalized corresponding image coordinates. The essential matrix is used to describe a relation between normalized image coordinates of two calibrated cameras and therefore their relative position and orientation.

$$\hat{x}'^T E \hat{x} = 0 \tag{14}$$

The Sampson error is checked for each point correspondence within the individual cameras. The essential matrix (Equation 15) of each individual camera, can be computed from the relative motion R and t as introduced in Equation 2

$$E = R[t]_x \tag{15}$$

where $[t]_x$ is the matrix representation of the cross product with t.

3.6 Refinement

In the paper two refinement techniques are utilized to enhance the results. I will only give a brief overview and will not go into detail since this step is not crucial for the general mechanics of the described approach and also would require a 3D point reconstruction of the scene to work. First non-linear refinement is used on all inliners, that are found using RANSAC. The cost function calculates the squared sum of deviations from a image point to a 3D world point, projected on to the image plane. The second enhancement step further improves the results with Kalman filtering. Therefore two independent 1D Kalman filters with constant velocity prior are used to smooth out noisy estimates for the yaw angle θ and the scale ρ. If at some time no point-correspondences can be found, the previous estimates of the last frame are propagated through the filters. In their real-world proof of concept implementation, they also used bundle adjustment and loop close techniques to improve the result.

4 Results

To evaluate the developed algorithm, a proof of concept data set was collected by using four fish-eye cameras that where already build in, to a commercially available car. Figure 6 shows the camera images used to collect the dataset. The cameras extrinsics were provided by the cars manufacturer and the cameras intrinsics were derived through calibration. To collect the dataset, they drove a 600m closed loop around a planar parking lot. During the drive, they captured a total of 4×2500 images and fed it into their algorithm. Besides that they collected GPS corrected inertial navigation system (GPS/INS) readings of the trajectory as ground truth data. The SURF feature extraction is done on the raw fish-eye images delivered by the cameras. For the non-linear refinement stage these images are undistorted using a fish-eye camera model in order to be able to do a efficient 3D triangulation needed for this stage.

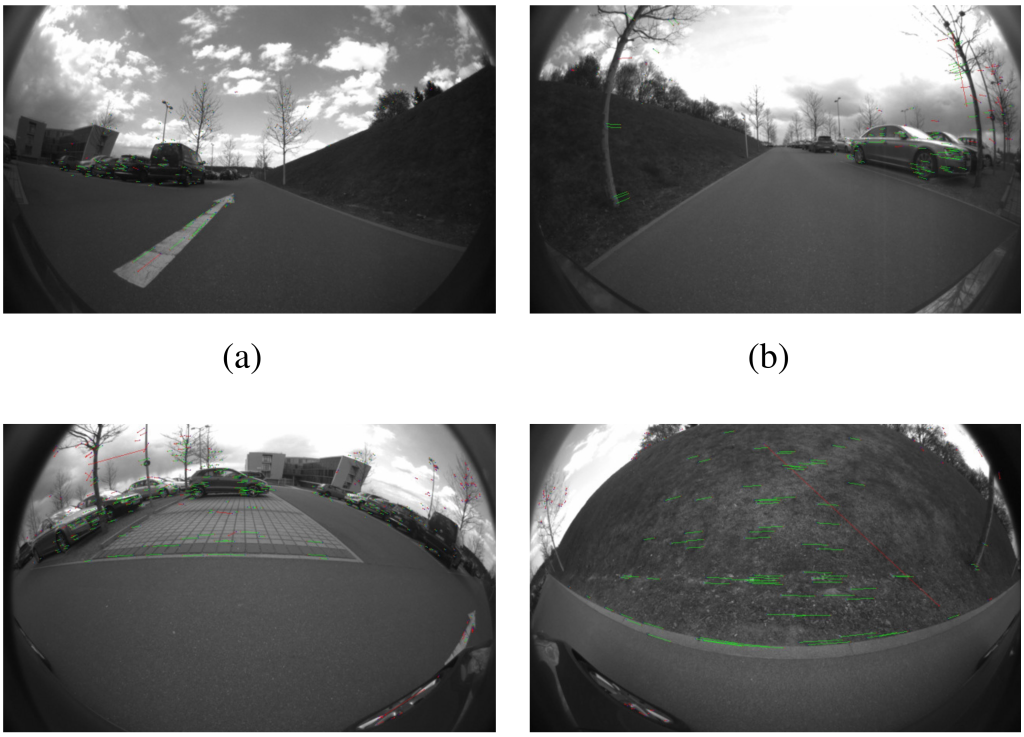


Figure 6: Images from the 4 cameras with fish-eye lens on the car used for testing. (a) Front, (b) Rear, (c) Left, (d) Right. (Source: [1])

Figure 8 shows the plots of all 2499 relative motions based on the first image as the reference frame. In Figure 8a, raw scale values recovered by the algorithm are compared to the ground truth and the filtered results. As it can be seen the scale estimations seem to be somewhat noisy, but show only a standard deviation of around 0.125m from GPS/INS. According to the authors the degenerated case when the car moves straight and one additional inter-camera correspondence is needed, was used 79.9% of the time. Figure 8b illustrates the recovered yaw angle of the car. As it can be seen, even without filtering the recovered angle is very close to the ground truth and shows no noticable drift. The fully recovered trajectory of the car can be seen in Figure 7.

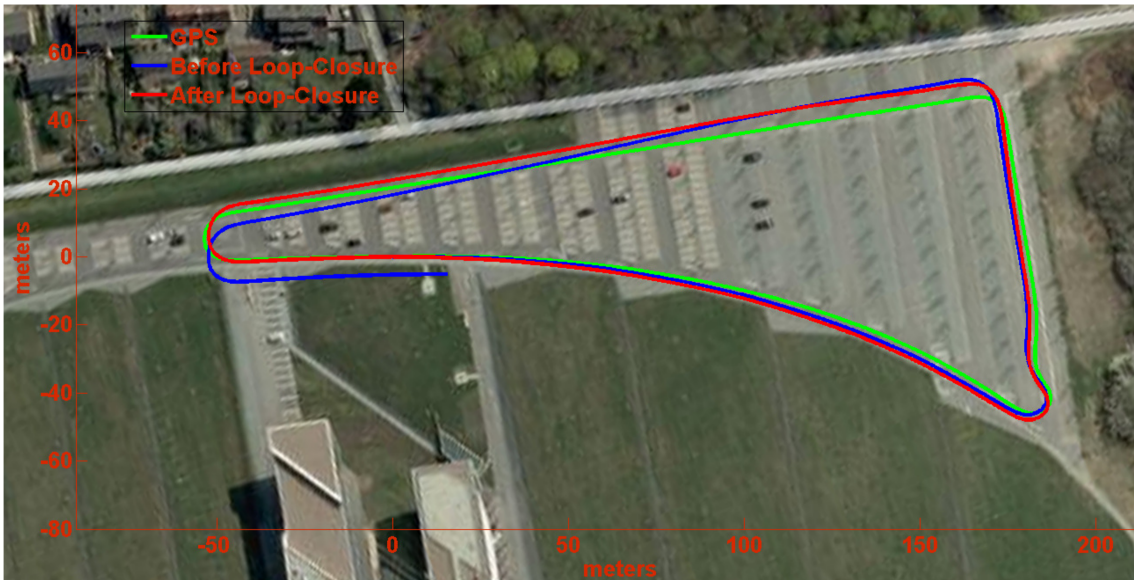
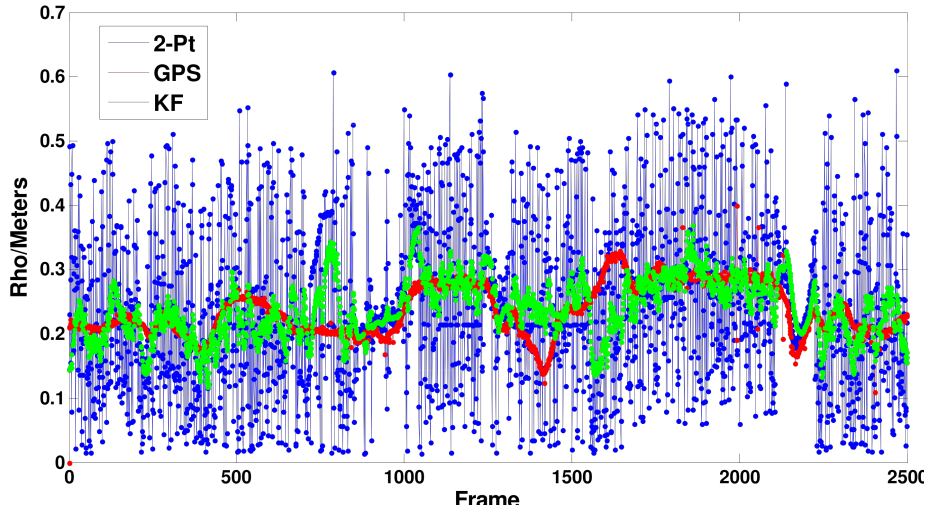
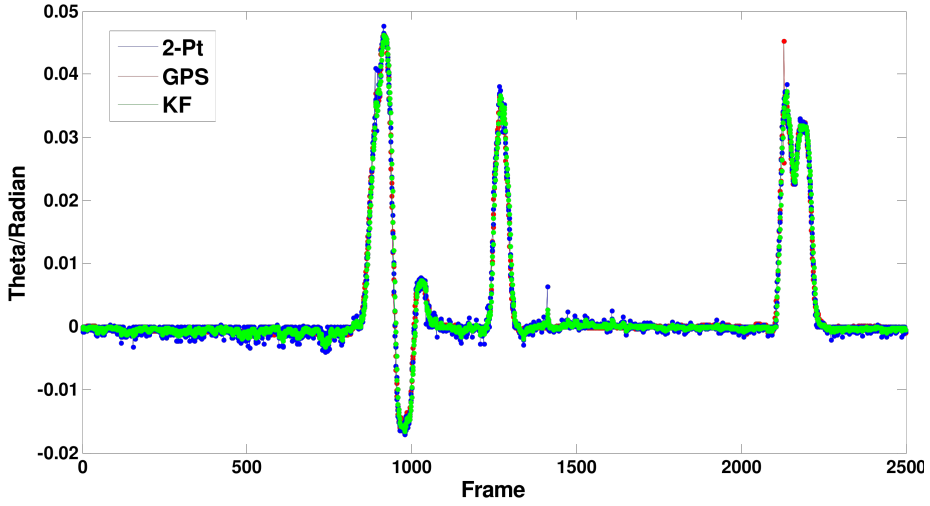


Figure 7: Top view of the recovered trajectory with enhancements compared to the GPS/INS ground truth. (Source: [1])



(a)



(b)

Figure 8: Scales ρ (a) and yaw angles θ (b) between all consecutive frames from the motion estimation algorithm compared to GPS/INS ground truth. (Source: [1])

5 Conclusions

The authors presented a new visual egomotion algorithm with metric scale for on-road vehicles. For their formulated model, they derived a analytical two point minimal solution for planar and circular motion. They identified a degeneracy in the model and supplied a suitable way to eliminate the degeneracy. With this they are able to calculate the relative translation and rotation of a moving vehicle based on camera images and hence recover its full trajectory. To demonstrate and evaluate their investigations, they created a proof of concept implementation and tested its practicality and robustness against degeneracy on real world data. Therefore they used a original sized car equipped with four cameras and collected real world data on a planar parking lot, driving in a closed loop.
Analyzing the collected data, they compared the computed raw data against different filters that smoothed the data and removed noise. The collected dataset contains 4×2500 images taken while driving a 600m long loop. At first this seems to be a fairly small dataset, but further investigations shows, that this size is comparable to the size used by other researchers. They compared the collected raw results against the filtered data and collected ground truth data using GPS. For the computed scales, they stated the standard deviation from ground truth and even analyzed how often inter-camera correspondences where used to compensate for the degeneracy.

According to the authors the intended real time requirements are met, because of their implementation being able of handling 6 frames per second. Besides that they are convinced, that the assumptions on vehicle motion made, hold for real world data. As shown by their real-world dataset the developed approach is able to recover the scale and the yaw angle successfully. Although recovery of the scale is somewhat noisy, the yaw angle can be calculated with high precision and no recognizable drift.
In my opinion, the real time capability with only 6 frames per second is just enough to deliver usable results, since it only allows for a rather rough resolution. A car driving around 60 kilometers per hour for example, would move around 2.8 meters between each frame. Besides that, a normal car does not really obey only planar circular motion. Still I think the developed approach was successful as they proved, that they can reduce the complexity. Their approach can be used in applications where planar motion is given and the results can be refined with additional sensor data to compute the scale. Because in their real-world test they had to use additional inter-camera correspondences around 80% of the time. With this they developed a base for further improvements and additional research. In the future investigations on how non-static scenes could affect the results should be done and larger datasets should be collected to establish a well-founded evaluation on the produced results.

6 Related Work

The authors gathered research done on different fields of egomotion and combined them to form a new and more advanced method. Pless [4] established the idea of using generalized camera systems where only one epipolar constraint for the whole system is used to describe relative motion, instead of one per camera pair. Pless derived the generalized essential matrix and introduced a algorithm for solving it. Later Li *et al.* [16] did work on enhancing the algorithm for solving for the generalized essential matrix and identified degeneracy when using intra-camera correspondences. Scaramuzza *et al.* [9] done research on using the Ackermann steering principle, to reduce the needed effort for solving for the generalized essential matrix to a 2-point algorithm. Since he was using omni-directional cameras he developed an extra approach to receive metric scale from it.

In [1] the authors combined this research to a new algorithm. Through combination of the generalized camera systems, the knowledge of degeneracy when using intra-camera and previous work on constraining motion using the Ackerann steering principle, they were able to derive a new solution. As Clipp *et al.* [17] has done it in the past, they used another camera to compensate for the degeneracy and retrieve scale. They were the first to show a 2-point algorithm for a generalized camera system which comes close to real time and they where the first demonstrating its functionality on real-world data.

References

[1] Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. "Motion Estimation for Self-Driving Cars With a Generalized Camera". In: *Computer Vision and Pattern Recognition* (2013).

[2] Martin John Baker. *Maths - Plücker Coordinates*. Feb. 3, 2016. URL: `http://www.euclideanspace.com/maths/geometry/elements/line/plucker/index.htm` (visited on 02/03/2016).

[3] Peter Sturm. "Multi-view geometry for general camera models". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 206–212.

[4] Robert Pless. "Using many cameras as one". In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2003, pp. II–587.

[5] Michael D Grossberg and Shree K Nayar. "A general imaging model and a method for finding its parameters". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. IEEE. 2001, pp. 108–115.

[6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[7] Julien Rabin Edouard Oyallon. *An Analysis of the SURF Method*. 2015. URL: `http://www.ipol.im/pub/art/2015/69/`.

[8] D.G. Lowe. *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image*. US Patent 6,711,293. Mar. 2004. URL: `http://www.google.com/patents/US6711293`.

[9] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. "Real-time monocular visual odometry for on-road vehicles with 1-point ransac". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 4293–4299.

[10] Chris Harris and Mike Stephens. "A combined corner and edge detector." In: *Alvey vision conference* 15 (1988), p. 50.

[11] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[12] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782. DOI: `10.1145/358669.358692`. URL: `http://doi.acm.org/10.1145/358669.358692`.

[13] Wikipedia. *Ackermann steering geometry — Wikipedia, The Free Encyclopedia*. [Online; accessed 02-March-2016]. 2016. URL: https://en.wikipedia.org/wiki/Ackermann_steering_geometry.

[14] Unknown. *The Sylvester Matrix and Resultants*. [Online; accessed 02-March-2016]. 2007. URL: http://math.rice.edu/~cbruun/vigre/vigreHW9.pdf.

[15] Masayuki Tanaka. *Sampson Error*. June 6, 2012. URL: http://like.silk.to/studymemo/SampsonError.pdf (visited on 02/23/2016).

[16] Hongdong Li, Richard Hartley, and Jae-Hak Kim. "A linear approach to motion estimation using generalized camera models". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[17] Brian Clipp et al. "Robust 6dof motion estimation for non-overlapping, multi-camera systems". In: *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*. IEEE. 2008, pp. 1–8.

Appendices

A (4) → (5)

The first big step from equation (4) → (5) in the paper, is to establish the coefficients obtained from a Plücker line correspondence. For reasons of clarity if will not present all the conducted sub-steps in the mathematical calculation. Lets denote a corresponding Plücker line pair $l = [u^T \quad (t_C \times u)^T]^T, l' = [u'^T \quad (t_{C'} \times u')^T]^T$ as followed.

$$\begin{pmatrix} u'_x \\ u'_y \\ u'_w \\ t_{C'y}u'_w - t_{C'w}u'_y \\ t_{C'w}u'_x - t_{C'x}u'_w \\ t_{C'x}u'_y - t_{C'y}u'_x \end{pmatrix} = \mathrm{l}' \qquad \begin{pmatrix} u_x \\ u_y \\ u_w \\ t_{Cy}u_w - t_{Cw}u_y \\ t_{Cw}u_x - t_{Cx}u_w \\ t_{Cx}u_y - t_{Cy}u_x \end{pmatrix} = \mathrm{l}$$

Next we fully expand the generalized epiporal constraint from left to right.

$$l'^T E_{GC}\, l = 0$$

$$l'^T \begin{bmatrix} 0 & 0 & \rho\ sin\ \frac{\theta}{2} & cos\ \theta & -sin\ \theta & 0 \\ 0 & 0 & -\rho\ cos\ \frac{\theta}{2} & sin\ \theta & cos\ \theta & 0 \\ \rho\ sin\ \frac{\theta}{2} & \rho\ cos\ \frac{\theta}{2} & 0 & 0 & 0 & 1 \\ cos\ \theta & -sin\ \theta & 0 & 0 & 0 & 0 \\ sin\ \theta & cos\ \theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} l = 0$$

$$\begin{pmatrix} u'_w\ \rho\ sin\frac{\theta}{2} + (t_{C'y}u'_w - t_{C'w}u'_y)cos\ \theta + (t_{C'w}u'_x - t_{C'x}u'_w)sin\ \theta, \\ u'_w\ \rho\ cos\frac{\theta}{2} + (t_{C'w}u'_x - t_{C'x}u'_w)cos\ \theta - (t_{C'y}u'_w - t_{C'w}u'_y)sin\ \theta, \\ u'_x\ \rho\ sin\frac{\theta}{2} - u'_y\ \rho\ cos\ \frac{\theta}{2} + (t_{C'x}u'_y - t_{C'y}u'_x), \\ u'_x cos\ \theta + u'_y sin\ \theta, \\ u'_y cos\ \theta - u'_x sin\theta, \\ u'_w \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ u_w \\ t_{Cy}u_w - t_{Cw}u_y \\ t_{Cw}u_x - t_{Cx}u_w \\ t_{Cx}u_y - t_{Cy}u_x \end{pmatrix} = 0$$

$$
\begin{aligned}
& u_x u'_w \ \rho \ sin\tfrac{\theta}{2} + u_x(t_{C'y}u'_w - t_{C'w}u'_y)cos\ \theta + u_x(t_{C'w}u'_x - t_{C'x}u'_w)sin\ \theta \\
+\ & u_y u'_w \ \rho \ cos\tfrac{\theta}{2} + u_y(t_{C'w}u'_x - t_{C'x}u'_w)cos\ \theta - u_y(t_{C'y}u'_w - t_{C'w}u'_y)sin\ \theta \\
+\ & u_w u'_x \ \rho \ sin\tfrac{\theta}{2} - u_w u'_y \ \rho \ cos\ \tfrac{\theta}{2} + u_w(t_{C'x}u'_y - t_{C'y}u'_x) \\
+\ & (t_{Cy}u_w - t_{Cw}u_y)u'_x cos\ \theta + (t_{Cy}u_w - t_{Cw}u_y)u'_y sin\ \theta \\
+\ & (t_{Cw}u_x - t_{Cx}u_w)u'_y cos\ \theta - (t_{Cw}u_x - t_{Cx}u_w)u'_x sin\ \theta \\
+\ & (t_{Cx}u_y - t_{Cy}u_x)u'_w = 0
\end{aligned}
$$

After that I rearrange the terms of the sum to match equation 5 of the paper in order to read out the coefficients of the Plücker line correspondences.

$$
\begin{aligned}
&\underbrace{\begin{bmatrix} u_x(t_{C'y}u'_w - t_{C'w}u'_y) + u_y(t_{C'w}u'_x - t_{C'x}u'_w) \\ +(t_{Cy}u_w - t_{Cw}u_y)u'_x + (t_{Cw}u_x - t_{Cx}u_w)u'_y \end{bmatrix}}_{\text{a}} cos\ \theta \\
&+ \underbrace{\begin{bmatrix} u_x(t_{C'w}u'_x - t_{C'x}u'_w) - u_y(t_{C'y}u'_w - t_{C'w}u'_y) \\ +(t_{Cy}u_w - t_{Cw}u_y)u'_y - u'_x(t_{Cw}u_x - t_{Cx}u_w) \end{bmatrix}}_{\text{b}} sin\ \theta \\
&+ \underbrace{\left[u_y u'_w - u_w u'_y\right]}_{\text{c}}\ \rho\ cos\ \tfrac{\theta}{2} \\
&+ \underbrace{\left[u_x u'_w + u_w u'_x\right]}_{\text{d}}\ \rho\ sin\tfrac{\theta}{2} \\
&+ \underbrace{u_w(t_{C'x}u'_y - t_{C'y}u'_x) + u'_w(t_{Cx}u_y - t_{Cy}u_x)}_{\text{e}} = 0
\end{aligned}
$$

Since the coefficients are differing from the paper we have to rearrange the equation step by step to get to the equation given in the paper. For that I will expand every coefficient ($a,\ b,\ c,\ d,\ e$) and reorganize its terms to isolate the wanted representation.

$$
\begin{aligned}
&\underbrace{\begin{bmatrix} u_y(t_{C'w}u'_x - t_{Cw}u'_x) + u_x(t_{Cw}u'_y - t_{C'w}u'_y) \\ -u_w(t_{Cx}u'_y - t_{Cy}u'_x) - u'_w(t_{C'x}u_y - t_{Cy}u_x) \end{bmatrix}}_{\text{a}} cos\ \theta \\
&+ \underbrace{\begin{bmatrix} u_x(t_{C'x}u'_w - t_{C'w}u'_x) - u_y(t_{C'y}u'_w - t_{C'w}u'_y) \\ -u_x(t_{Cx}u_w - t_{Cw}u_x) - u'_y(t_{Cy}u_w - t_{Cw}u_y) \end{bmatrix}}_{\text{b}} sin\ \theta \\
&+ \underbrace{\left[u_w u'_y - u_y u'_w\right]}_{\text{c}}\ \rho\ cos\ \tfrac{\theta}{2} \\
&+ \underbrace{\left[u_x u'_w + u_w u'_x\right]}_{\text{d}}\ \rho\ sin\tfrac{\theta}{2} \\
&+ \underbrace{u_w(t_{C'x}u'_y - t_{C'y}u'_x) + u'_w(t_{Cx}u_y - t_{Cy}u_x)}_{\text{e}} = 0
\end{aligned}
$$

B $(8,\ 9) \to (10)$

Once we obtained the coefficients listed in the paper, the next big step from equation $(8, 9) \to (10)$ in the paper, is to obtain the coefficients $A,\ B,\ C,\ D$ using the Sylvester Resultant method. For reasons of clearity if will not present all the conducted sub-steps in the mathematical calculation. The resultant of the two polynomials from $(8,\ 9)$ is a six degrees polynomial equation equal to the determinant of the sylvester matrix. To calculate the Sylvester Resultant, we just have to establish the sylvester matrix and calculate its determinant.

$$\begin{bmatrix} (2\,a_1\,d_2 2\,b_1\,c_2 + 2\,c_1\,b_2 + 2\,d_1\,a_2)\ \beta^3 + (d_1\,e_2 + (e_1 + a_1) & & \\ d_2 + 2\,b_1\,c_2 2\,c_1\,b_2 d_1\,a_2\,\beta & (2\,b_1\,d_2 + 2\,a_1\,c_2 + 2\,d_1\,b_2 2\,c_1\,a_2)\ \beta^4 + (c_1\,e_2 + 2\,b_1\,d_2 + (e_1 3\,a_1)\ c_2 2\,d_1\,b_2 + 3\,c_1\,a_2)\ \beta^2 c_1\,e_2 + (e_1 + a_1)\ c_2 c_1\,a_2 \\ (2\,b_1\,d_2 2\,a_1\,c_2 2\,d_1\,b_2 + 2\,c_1\,a_2)\ \beta^2 c_1\,e_2 + (e_1 + a_1)\ c_2 c_1\,a_2 & (2\,a_1\,d_2 2\,b_1\,c_2 + 2\,c_1\,b_2 + 2\,d_1\,a_2)\ \beta^3 + (d_1\,e_2 + (e_1 + a_1)\ d_2 + 2\,b_1\,c_2 2\,c_1\,b_2 d_1\,a_2)\ \beta \end{bmatrix}$$

Calculating the determinant of the matrix leaves us with equation (10a) of the paper

$$A\beta^6 + B\beta^4 + C\beta^2 + D = 0$$

where its coefficients $A,\ B,\ C,\ D$ are

$$\begin{aligned}
A = {} & \left(4b_1^2 + 4a_1^2\right)\ d_2^2 + \left(\left(8\,b_1\,d_1 8\,a_1\,c_1\right)\ b_2 + \left(8\,b_1\,c_1 8\,a_1\,d_1\right)\ a_2\right)\ d_2 \\
& + \left(4\,b_1^2 + 4\,a_1^2\right)\ c_2^2 + \left(\left(8\,a_1\,d_1 8\,b_1\,c_1\right)\ b_2 + \left(8\,b_1\,d_1 8\,a_1\,c_1\right)\ a_2\right)\ c_2 \\
& + \left(4\,d_1^2 + 4\,c_1^2\right)\ b_2^2 + \left(4\,d_1^2 + 4\,c_1^2\right)\ a_2^2 \\
B = {} & \left(\left(4\,a_1\,d_1 4\,b_1\,c_1\right)\ d_2 + \left(4\,b_1\,d_1 + 4\,a_1\,c_1\right)\ c_2 + \left(4\,d_1^2 4\,c_1^2\right)\ a_2\right)\ e_2 \\
& + \left(4\,a_1\,e_1 4\,b_1^2 4\,a_1^2\right)\ d_2^2 + \left(\left(4\,c_1\,e_1 + 8\,b_1\,d_1 + 12\,a_1\,c_1\right)\ b_2 + \left(4\,d_1\,e_1 + 8\,a_1\,d_1 12\,b_1\,c_1\right)\ a_2\right)\ d_2 \\
& + \left(4\,a_1\,e_1 8\,b_1^2 8\,a_1^2\right)\ c_2^2 + \left(\left(4\,d_1\,e_1 12\,a_1\,d_1 + 16\,b_1\,c_1\right)\ b_2 + \left(4\,c_1\,e_1 + 12\,b_1\,d_1 + 16\,a_1\,c_1\right)\ a_2\right)\ c_2 \\
& + \left(4\,d_1^2 8\,c_1^2\right)\ b_2^2 + \left(4\,d_1^2 8\,c_1^2\right)\ a_2^2 \\
C = {} & \left(d_1^2 + c_1^2\right)\ e_2^2 + \left(\left(2\,d_1\,e_1 2\,a_1\,d_1 + 4\,b_1\,c_1\right)\ d_2 + \left(2\,c_1\,e_1 4\,b_1\,d_1 6\,a_1\,c_1\right)\ c_2 + \left(2\,d_1^2 + 6\,c_1^2\right)\ a_2\right)\ e_2 \\
& + \left(e_1^2 + 2\,a_1\,e_1 + a_1^2\right)\ d_2^2 + \left(\left(4\,c_1\,e_1 4\,a_1\,c_1\right)\ b_2 + \left(2\,d_1\,e_1 2\,a_1\,d_1 + 4\,b_1\,c_1\right)\ a_2\right)\ d_2 \\
& + \left(e_1^2 + 6\,a_1\,e_1 + 4\,b_1^2 + 5\,a_1^2\right)\ c_2^2 + \left(\left(4\,d_1\,e_1 + 4\,a_1\,d_1 8\,b_1\,c_1\right)\ b_2 + \left(6\,c_1\,e_1 4\,b_1\,d_1 10\,a_1\,c_1\right)\ a_2\right)\ c_2 \\
& + 4\,c_1^2\,b_2^2 + \left(d_1^2 + 5\,c_1^2\right)\ a_2^2 \\
D = {} & c_1^2\,e_2^2 + \left(\left(2\,c_1\,e_1 + 2\,a_1\,c_1\right)\ c_2 2\,c_1^2\,a_2\right)\ e_2 \\
& + \left(e_1^2 2\,a_1\,e_1 a_1^2\right)\ c_2^2 + \left(2\,c_1\,e_1 + 2\,a_1\,c_1\right)\ a_2\,c_2 c_1^2\,a_2^2
\end{aligned}$$

Since D does not look like equation (11) of the paper, we once again have to expand and rearange it.

$$\begin{aligned}
D &= c_1^2\,e_2^2 + \left(\left(2\,c_1\,e_1 + 2\,a_1\,c_1\right)\ c_2 2\,c_1^2\,a_2\right)\ e_2 \\
&\quad + \left(e_1^2 2\,a_1\,e_1 a_1^2\right)\ c_2^2 + \left(2\,c_1\,e_1 + 2\,a_1\,c_1\right)\ a_2\,c_2 c_1^2\,a_2^2 \\
&= \underline{-c_2^2(e_1^2 + a_1^2) - 2c_2^2 e_1 a_1 - c_1^2(e_2^2 + a_2^2) - 2c_1^2 e_2 a_2} \\
&\quad \underline{+2c_2 c_1(a_1 a_2 + e_1 e_2) + 2c_2 c_1(e_1 a_2 + a_1 e_2)}
\end{aligned}$$