Using Polynomial Eigenvalue Problem Modeling to Improve Visual Odometry for Autonomous Vehicles

Anderson Souza[1], Leonardo Souto[2], Fabio Fonseca de Oliveira[2], Biswa Nath Datta[3] and Luiz M. G. Gonçalves[2]

[1] *Department of Computing, University of the State of Rio Grande do Norte, Natal, RN, Brazil*

[2] *Department of Computing Engineering and Automation, Federal University of the Rio Grande do Norte, Natal, RN, Brazil*

[3] *Northern Illinois University, U.S.A.*

Keywords: Visual Odometry, Polynomial Eigenvalue Problem, Motion Estimation.

Abstract: Visual Odometry (VO) is the process of calculating the motion of an agent (such as, robot and vehicle), using images captured by a single or multiple cameras embedded to it. VO is an important process to supplement autonomous navigation systems, since VO can provide accurate trajectory estimates. However, algorithms of VO work with several steps of hard numerical computation which generate numerical errors and demand considerable processing time. In this paper, we propose the use of a mathematical framework for monocular VO process based on Polynomial Eigenvalue Problem (PEP) modeling in order to achieve both more accurate motion estimation and to decrease the processing time of the VO process. Some previous experiments are shown in order to validate the proposed computation accuracy.

1 INTRODUCTION

Nowadays, it is common to see autonomous agents such as robot and vehicle to perform different tasks. It is possible to find out these devices transporting materials in factories, transporting people in urban zone, monitoring environments, exploring areas, in surveillance, among other applications (Siegwart et al., 2011).

Autonomous agents must be able to collect information about their environment; considering this information, they must make some decision in order to decide how to proceed, facing what exists in the environment; and have to actuate in order to perform the previous decision, towards completing their mission (Murphy, 2000). An important condition for agents achieving these abilities is that they must be equipped with sensors, which provide them useful information of their environment. In this way, a robot or a vehicle can interact coherently with its environment and objects, leading with unexpected situations like, dynamic obstacles (Souza and Gonçalves, 2015).

Cameras are widely used as visual sensor systems for autonomous robots and vehicles. These systems can be composed by a single or multiples cameras and a mechanism for computing cameras data, which allows the extraction of useful information from raw data (Ma et al., 2004). With a visual sensor system it is possible to infer a plenty of information such as colors, textures, geometric structures, object recognition, among others. Furthermore, it is feasible to estimate relative or absolute motion, from images captured from different positions, *Visual Odometry - VO*. This is an important operation to supplement autonomous navigation systems, since VO can provide accurate trajectory estimates (Scaramuzza and Fraundorfer, 2011).

However, processing raw data captured by cameras in order to extract useful information, goes through several numerical computation steps, which produce numerical errors (overflow and underflow), round-off errors and error bound. Moreover, these computations consume significant processing time demanded by algebraic calculations with vectors and matrices (Datta, 2010). Both aspects (errors effects and time processing) need to be minimized so that they do not cause damages to the autonomous navigation process by misinterpreted information (due to miscalculations), or through decisions not taken in time.

In this context, this paper proposes the use of a mathematical framework for monocular VO process

Souza, A., Souto, L., Oliveira, F., Datta, B. and Gonçalves, L.
Using Polynomial Eigenvalue Problem Modeling to Improve Visual Odometry for Autonomous Vehicles.
DOI: 10.5220/0006478005020507
In *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2017) - Volume 2*, pages 502-507
ISBN: Not Available

based on Polynomial Eigenvalue Problem (PEP) modeling in order to achieve both more accurate motion estimation and decreasing the processing time of the VO process. In this way, the VO process can be analyzed as a minimum problem and it can be solved by using polynomial equations systems (Kukelova et al., 2012). In order to validate the proposed modeling in terms of accuracy, preliminary experiments with real data are presented.

This paper is organized as follow: section 2 introduces the Visual Odometry problem for a monocular vision system. Section 3 provides the formulation of PEP. Section 4 shows the Visual Odometry modeled as a PEP. Section 5 presents some preliminary experiments. Finally, some considerations and future works are presented in section 6.

2 MONOCULAR VISUAL ODOMETRY

Localization is a fundamental point in autonomous navigation. An autonomous robot or vehicle needs to be able to estimate its position and orientation relative to its environment. Visual Odometry - VO is process that can provide accurate trajectory estimates. VO operates by incrementally estimating the pose (position and orientation) of an agent through examination of the changes that motion induces on images of its onboard cameras (Scaramuzza and Fraundorfer, 2011).

VO process can be performed by multiples cameras or a single camera, which is named monocular VO. In this case, motion and 3D structures are computed from 2D images data. Feature-based methods is one of the main approaches for estimating poses in monocular VO. These methods are based on salient and repeatable features that are tracked over consecutive frames (Scaramuzza and Fraundorfer, 2011).

VO computes the camera path incrementally, pose after pose (for simplicity, it is common to assume that the camera coordinate frame to be the agent coordinate frame). In order to do this, the main point is to compute the relative transformation $T_k \in \mathbb{R}^{4\times4}$, $(k = 1,\dots,n)$, from consecutive images (I_{k-1}, I_k) taken at times $k-1$ and k and then, concatenate the transformation to recover the full trajectory $C_{0:n} = \{C_0, C_1, \dots, C_n\}$. Figure 1 illustrates this process. T_k is often called rigid body motion and can be described as:

$$T_k = \begin{bmatrix} R_k & t_k \\ 0 & 1 \end{bmatrix} \tag{1}$$

$R_k \in SO(3)$ and $t_k \in \mathbb{R}^{3\times1}$ are the rotation matrix and the translation vector, respectively. These parameters can be computed by estimating the essential matrix E, which describes the geometric relation between two images I_k and I_{k-1} of a calibrated camera, up to a scale factor, as shown in Equation (2).

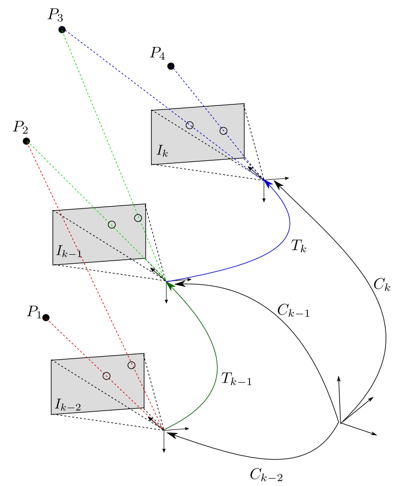

Figure 1: The relative motion T_k is calculated from features (projections of P_2 and P_3) and after, it is concatenated to obtain the absolute pose C_k.

$$E_k \simeq \hat{t}_k R_k \tag{2}$$

where $t_k = [t_x, t_y, t_z]^T$ and

$$\hat{t}_k = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \tag{3}$$

It is known that in the case of fully calibrated cameras points q and q' (from I_{k-1} and I_k, respectively), which are projections of 3D points P (Figure 1), are geometrically constrained by the epipolar geometry constraint (Kukelova et al., 2012), formulated by Equation (4).

$$q' E_k q = 0 \tag{4}$$

where the essential matrix E_k is a 3×3 rank-2 matrix with two equal singular values. These constraints can be presented in an other way, as shown in Equations (5) and (6).

$$det(E_k) = 0 \tag{5}$$

$$2E_k E_k^T E_k - trace(E_k E_k^T) E_k = 0 \tag{6}$$

Five-points algorithm proposed by Nister (Nister, 2004) and eight-points algorithm proposed by Longuet-Higgins (Longuet-Higgins, 1981) are the most popular approaches for computing the essential matrix E_k and, recovering R_k and t_k. In this paper, we focus on the polynomial eigenvalue solution to the five-points algorithm for estimating E_k.

Usually, after getting a possible result of E_k, a nonlinear optimization step is performed in order to obtain more accurate estimate of T_k.

The VO process is often summarized into some steps as shown in Figure 2. Further information about VO can be found in (Scaramuzza and Fraundorfer, 2011) and (Scaramuzza and Fraundorfer, 2012), including mathematical formulation of VO problem, feature selection methods, matching, robustness and optimization approaches and applications.

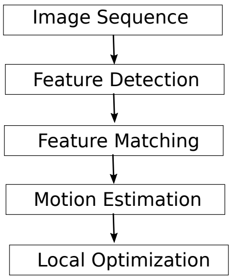

Figure 2: Steps of the monocular feature-based VO process.

3 POLYNOMIAL EIGENVALUE PROBLEMS

Following definitions and formulation presented in (Kukelova et al., 2012) and (Betcke et al., 2013), polynomial eigenvalue problems (PEP) are problems of the form presented by Equation (7), in which the main purpose is to find scalars λ and nonzero vectors $\mathbf{v}$ that satisfy the equation.

$$C(\lambda)\mathbf{v} = 0 \tag{7}$$

In this equation, $\mathbf{v}$ is vector of monomials in all variables except for λ, and $C(\lambda)$ is a $n \times n$ matrix polynomial in variable λ defined as

$$C(\lambda) \equiv \lambda^l C_l + \lambda^{l-1} C_{l-1} + \ldots + \lambda C_1 + C_0, \tag{8}$$

with $n \times n$ coefficient matrices C_j.

A polynomial eigenvalue problem (PEP) can be also represented as a standard generalized eigenvalue problem (GEP) with the form

$$A\mathbf{x} = \lambda B\mathbf{y} \tag{9}$$

Consider the following polynomial equation (PEP),

$$(\lambda^l C_l + \lambda^{l-1} C_{l-1} + \ldots + \lambda C_1 + C_0)\mathbf{v} = 0 \tag{10}$$

It can be transformed to a GEP with,

$$A = \begin{bmatrix} 0 & I & 0 & \ldots & 0 \\ 0 & 0 & I & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -C_0 & -C_1 & -C_2 & \ldots & -C_{l-1} \end{bmatrix}, \tag{11}$$

$$B = \begin{bmatrix} I & & & \\ & \ldots & & \\ & & I & \\ & & & C_l \end{bmatrix}, \mathbf{y} = \begin{bmatrix} \mathbf{v} \\ \lambda\mathbf{v} \\ \ldots \\ \lambda^{l-1}\mathbf{v} \end{bmatrix}$$

If C_l is nonsingular and well conditioned, it is possible consider a monic matrix polynomial,

$$\overline{C(\lambda)} = C_l^{-1} C(\lambda) \tag{12}$$

with coefficient matrices $\overline{C}_i = C_l^{-1} C_i$, $i = 0, \ldots, l-1$. Thus, Equation (10) can be transformed to

$$A\mathbf{y} = \lambda\mathbf{y}, \tag{13}$$

where

$$A = \begin{bmatrix} 0 & I & 0 & \ldots & 0 \\ 0 & 0 & I & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -\overline{C}_0 & -\overline{C}_1 & -\overline{C}_2 & \ldots & -\overline{C}_{l-1} \end{bmatrix} \tag{14}$$

In some cases, matrix C_l is singular, in contrast the matrix C_0 is regular and well conditioned. Thus, either the described method which transforms PEP (10) to the GEP (11) or the transformation $\beta = 1/\lambda$ can be used. Then, $\overline{C}_i = C_0^{-1} C_i$, $i = 1 \ldots l$ and matrix A gets the form

$$A = \begin{bmatrix} 0 & I & 0 & \ldots & 0 \\ 0 & 0 & I & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ -\overline{C}_l & -\overline{C}_{l-1} & -\overline{C}_{l-2} & \ldots & -\overline{C}_1 \end{bmatrix} \tag{15}$$

The transformation $\beta = 1/\lambda$ reduces the problem to finding eigenvalues of the matrix (Kukelova et al., 2012). There are many efficient numerical algorithms for solving this GEP, like the QZ algorithm (Datta, 2010).

4 PEP SOLUTION FOR VISUAL ODOMETRY

As mentioned before, five-points algorithm (Nister, 2004) is one of the most common approach for the

Visual Odometry problem. It has become the standard for monocular motion estimation in the presence of outliers (Scaramuzza and Fraundorfer, 2011).

Kukelova and colleagues (Kukelova et al., 2012) propose a solution based on PEP for relative pose problems using part of the five-points algorithm, and their formulation is used in this paper for the monocular visual odometry problem.

The five-points algorithm requires $m \geq 5$, with m been the number of points with correspondence in two consecutive images. Each of the point correspondences gives rise to a constraint of the form (4). This constraint can also be written as

$$\tilde{q}^T \tilde{E} = 0 \tag{16}$$

where

$$\tilde{q} \equiv [q_1 q'_1, q_2 q'_1, q_3 q'_1, q_1 q'_2, q_2 q'_2, q_3 q'_2, q_1 q'_3, q_2 q'_3, q_3 q'_3]^T$$

$$\tilde{E} = [e_{11}, e_{12}, e_{13}, e_{21}, e_{22}, e_{23}, e_{31}, e_{32}, e_{33}]^T$$

By stacking the vectors $\tilde{q}^T$ for five points, a 5×9 matrix is obtained. The null space of this matrix generates four vectors $\tilde{E_1}$, $\tilde{E_2}$, $\tilde{E_3}$ and $\tilde{E_4}$ that span it. They can be found by SVD or QR decomposition. With these vectors one can create four correspondent 3×3 matrices, E_1, E_2, E_3 and E_4. The essential matrix E can be constructed as a linear combination of these matrices, as shown by Equation (17).

$$E = xE_1 + yE_2 + zE_3 + E_4 \tag{17}$$

for some scalars x, y and z. Now, it is used the rank constraint (5) and the trace constraint (6) to build 10 third-order polynomial equations in three unknowns and 20 monomials. These equations can be written as:

$$MX = 0 \tag{18}$$

where M is a 10 coefficient matrix reduced by Gauss-Jordan elimination and $X = [x^3, yx^2, y^2x, y^3, zx^2, zyx, zy^2, z^2x, z^2y, z^3, x^2, yx, y^2, zx, zy, z^2, x, y, z, 1]^T$ is the vector of all monomials. There are all monomials in three unknowns up to degree three. At this point, Equation (10) must be recovered, and taking $\lambda = z$, Equation (17) can be written as

$$(z^3C_3 + z^2C_2 + zC_1 + C_0)\mathbf{v} = 0 \tag{19}$$

where $\mathbf{v}$ vector of monomials, $\mathbf{v} = [x^3, x^2y, xy^2, y^3, x^2, xy, y^2, x, y, 1]^T$ and C_3, C_2, C_1, and C_0 are 10×10 coefficient matrices given by:

$C_3 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ m_{10}]$,
$C_2 = [0\ 0\ 0\ 0\ 0\ 0\ 0\ m_8\ m_9\ m_{16}]$,
$C_1 = [0\ 0\ 0\ 0\ m_5\ m_6\ m_7\ m_{14}\ m_{15}\ m_{19}]$,
$C_0 = [m_1\ m_2\ m_3\ m_4\ m_{11}\ m_{12}\ m_{13}\ m_{17}\ m_{18}\ m_{20}]$.

where m_j is the jth column from M.

The rank of the matrix C_3 is one and the matrix C_0 is regular. Then, the transformation $\beta = 1/z$, presented in section 3, is possible and it reduces the cubic PEP (19) to the problem of finding the eigenvalues of the 30×30 matrix A, Equation (20).

$$A = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -c_0^{-1}C_3 & -C_0^{-1}C_2 & -C_0^{-1}C_1 \end{bmatrix} \tag{20}$$

From (20) 30 eigenvalues can be obtained, solutions for $\beta = 1/z$, and 30 corresponding eigenvectors $\mathbf{v}$ from which the solutions for x and y is extracted. Hence, the essential matrix can be estimated by Equation (17).

After that, R and t are recovered. Let the singular value decomposition of the essential matrix be $E \sim Udiag(1,1,0)V^T$, where U and V are chosen such that $det(U) > 0$ and $det(V) > 0$. Then, $t_u \equiv [u_{13}, u_{23}, u_{33}]^T$ and R is equal to $R_a \equiv UDV^T$ or $R_b \equiv UD^TV^T$ (Nister, 2004).

Any combination of R and t satisfies the epipolar constraint (16). Therefore, four possible solutions for the transformation T_k arise:

$$T_k^a = \begin{bmatrix} R_a & t_u \\ 0 & 1 \end{bmatrix}, T_k^b = \begin{bmatrix} R_a & -t_u \\ 0 & 1 \end{bmatrix}$$

$$T_k^c = \begin{bmatrix} R_b & t_u \\ 0 & 1 \end{bmatrix}, T_k^d = \begin{bmatrix} R_b & -t_u \\ 0 & 1 \end{bmatrix}$$

The true configuration is found by triangulating one point of the images for one of each possible solution and verifying if its coordinate in the space yield a position in front of the camera. Further details of the triangulation step can be found in (Nister, 2004).

5 PRELIMINARY EXPERIMENTS

This section shows some preliminary steps implemented to validate the proposed framework of VO computation in terms of accuracy.

5.1 Image Capturing and Feature Detection

For this first experiments, the KITTI dataset (Geiger et al., 2012) was used to provide the sequence of images and the ground truth of a real vehicle. The FAST (Features from Accelerated Segment Test) (Rosten et al., 2010) algorithm was applied on the images in order to identifies corners in consecutive images.

Figures 3 and 4 show features detected in consecutive images from the KITTI dataset.

Figure 3: Corners detected from the first image taken at time $k-1$.

Figure 4: Corners detected from the second image taken at time k.

5.2 Feature Matching

In this step was used the FLANN (Fast Library for Approximate Nearest Neighbors) algorithm (Muja and Lowe, 2009), which matches the detected corners. Figure 5 illustrates a matching sampling.

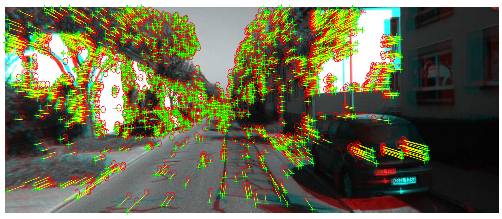

Figure 5: Matching example.

5.3 VO Experiment

An experiment was performed in order to compare the traditional implementation of VO using the Nister's five-point algorithm (Nister, 2004) and the implementation based on PEP with the modeling proposed by Kukelova and colleagues (Kukelova et al., 2012).

Figure 6 presents a monocular VO process performed with a set of 400 images from KITTI dataset. The figure shows that the PEP solution is closer to the ground truth line, indicating that the PEP solution is more accurate than the VO based on the Nister's five-point algorithm. This happened due to numerical robustness yield by the PEP solution.

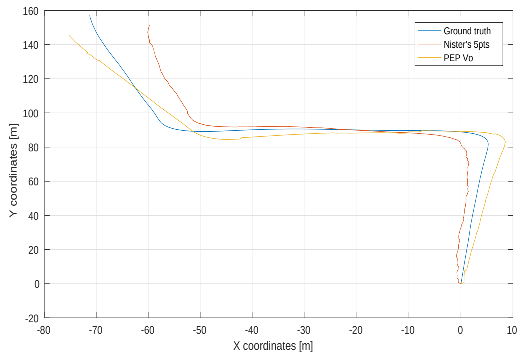

Figure 6: Comparative experiment with VO based on Nister's five-points algorithm and PEP solution.

6 CONCLUSION

This paper presented Monocular Visual Odometry (VO) computing solution based on Polynomial Eigenvalue Problem for the five-points algorithm. VO algorithms work with several step of hard numerical computation which generate numerical errors like overflow, underflow, round-off errors and error bound, and demand considerable processing time. VO process based on Polynomial Eigenvalue Problem (PEP) achieved more accurate motion estimation, since PEP solutions achieve more numerical robustness, as shown in a previous experiment.

As future works, the proposed method will be improved with more robust optimization step and new analysis comparing accuracy and processing will be performed.

The final idea is to apply the VO algorithm in a quadcopter vehicle in order to supply it with a estimation of its trajectory.

ACKNOWLEDGEMENTS

We would like to thank CAPES for the financial support.

REFERENCES

Betcke, T., Higham, N. J., Mehrmann, V., Schröder, C., and Tisseur, F. (2013). Nlevp: A collection of nonlinear eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 39(2).

Datta, B. (2010). *Numerical Linear Algebra and Applications*. SIAM Publishing, second edition edition.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kukelova, Z., Bujnak, M., and Pajdla, T. (2012). Polynomial eigenvalue solutions to minimal problems in computer vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 34(7):1381–1391.

Longuet-Higgins, H. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(10):133–135.

Ma, Y., Soatto, S., Kosecka, J., and Sastry, S. (2004). *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer.

Muja, M. and Lowe, D. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application*.

Murphy, R. (2000). *Introduction to AI Robotics*. MIT Press.

Nister, D. (2004). An efficient solution to the five-point relative pose. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Rosten, E., Porter, R., and Drummond, T. (2010). Faster and better: a machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry - part i: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):1–18.

Scaramuzza, D. and Fraundorfer, F. (2012). Visual odometry - part ii: Matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, 19(2):78–90.

Siegwart, R., Nourbakhsh, I., and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT Press, 2nd edition.

Souza, A. and Gonçalves, L. M. G. (2015). Occupancy-elevation grid: An alternative approach for robotic mapping and navigation. *Robotica, Cambridge University Press*, pages 1–18.