Image and Vision Computing 50 (2016) 14–26

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Estimating the focus of expansion in a video sequence using the trajectories of interest points☆

Pedro Gil-Jiménez*, Hilario Gómez-Moreno, Roberto J. López-Sastre, Alberto Bermejillo-Martín-Romo

Dpto. de Teoría de la Señal y Comunicaciones, Universidad de Alcalá, 28805, Alcalá de Henares, (Madrid), Spain

ARTICLE INFO

Article history:
Received 2 July 2015
Received in revised form 22 December 2015
Accepted 23 March 2016
Available online 31 March 2016

Keywords:
Focus of expansion
Optical flow
Interest point
Point trajectory
Cross ratio

ABSTRACT

In this paper, we present a new algorithm for the computation of the focus of expansion in a video sequence. Although several algorithms have been proposed in the literature for its computation, almost all of them are based on the optical flow vectors between a pair of consecutive frames, so being very sensitive to noise, optical flow errors and camera vibrations. Our algorithm is based on the computation of the vanishing point of point trajectories, thus integrating information for more than two consecutive frames. It can improve performance in the presence of erroneous correspondences and occlusions in the field of view of the camera. The algorithm has been tested with virtual sequences generated with Blender, as well as some real sequences from both, the public KITTI benchmark, and a number of challenging video sequences also proposed in this paper. For comparison purposes, some algorithms from the literature have also been implemented. The results show that the algorithm has proven to be very robust, outperforming the compared algorithms, specially in outdoor scenes, where the lack of texture can make optical flow algorithms yield inaccurate results. Timing evaluation proves that the proposed algorithm can reach up to 15fps, showing its suitability for real-time applications.

1. Introduction

When a camera moves across a rigid scene, the apparent motion of the imaged points can be used to infer the relative shift of the camera with respect to the scene. For the general case, the problem consists in the computation of the translational and rotational vectors, and is called *ego-motion* [1]. The computation of ego-motion plays an important role in some vision systems, such as Visual Odometry, 3D reconstruction, time-to-impact estimation or obstacle detection and avoidance.

When the rotational component is null, that is, the camera moves in a straight line, the problem reduces to the computation of the translational vector, and the image of this vector on the image plane is called the *Focus of Expansion* (FoE) when the camera moves forwards, or the *Focus of Contraction* (FoC) when it moves backwards, see Fig. 1. Although FoE and FoC refer to opposite directions, their properties are very similar, and we will only refer to the former, recalling that the differences in the computation of the later are minimal.

For its computation, many algorithms have been proposed. In the classic approach, the focus of expansion is computed from the optical flow field between two time-varying frames, which can be obtained from several algorithms [2]. However, technical challenges still exist for general scenes. Typical inaccuracies raise in unconstrained environments, such as road scenes where a large proportion of the image appears untextured, for instance, the sky or a textureless pavement, and optical flow vectors for these areas do not exist or are erroneous. Another source of error can be caused by vibrating platforms. Although many vision benchmarks are available for research, such as the *KITTI Benchmark* [3] or the *CMU Visual Localization Data Set* [4], these sequences are recorded with complex camera setups to prevent such problems. For more basic setups, however, any instability can cause the FoE computation to decrease its accuracy.

To face these problems, we propose a new method based on the estimation of the vanishing point for multi-frame interest point trajectories. The paper is structured as follows: In Section 2, a brief overview of different FoE estimation approaches is provided. In Section 3 we introduce our algorithm for the FoE computation based on the trajectories of interest points. A comparison of our algorithm with other works is given in Section 4, while Section 5 concludes the paper.

☆ This paper has been recommended for acceptance by Cornelia M Fermuller.

* Corresponding author.
E-mail address: pedro.gil@uah.es (P. Gil-Jiménez).

http://dx.doi.org/10.1016/j.imavis.2016.03.007

2. Related works

Existing techniques to estimate the FoE can be grouped into two main approaches, namely the *continuous methods* and the *discrete methods*. Algorithms from the continuous group employ dense optical flow fields, as in [5–7]. In [8], Sazbon et al. recall that, for a camera moving in a rigid scene, the FoE is characterized by a null flow vector, with the optical flow field radially diverging from it. Thus, only the angular component is enough for the estimation of the FoE, ignoring the magnitude component of the optical flow field. In their work, it is proposed the use of a specially designed matched filter which can work with a low-quality estimation of the optical flow. The matched filter is used to refine the FoE location after a rough estimation in a first phase. However, the algorithm proposed in this work requires that the flow estimation strongly covers the area near the FoE, but this is not generally the case for general video sequences.

The main disadvantage of continuous methods is that dense optical flow is computationally expensive. Furthermore, scenes with lack of texture on a large proportion of the image can yield inaccurate results, since optical flow fields for these areas are likely to be erroneous. To solve these problems, discrete methods use stronger correspondences between image features, such as points or lines, that can be computed from sparse optical flow algorithms.

If several correspondences between points from two consecutive frames are available, the *Fundamental Matrix* F can be computed, and the FoE will correspond with the null-vector of F [9] (p. 245). Being the basic approach for the FoE estimation, the Fundamental Matrix is very inaccurate, and has not been widely used. In [10], the essential matrix is used instead. However, the results show that the iterative Levenberg–Marquardt algorithm is needed to improve results from linear algorithms, increasing thus the computational complexity.

Another approach consists in the computation of the intersection point of all the lines defined by the optical flow vectors. Since noise makes all the lines not intersecting at the same point, a minimization criterion is needed. In [11], Suhr et al. accumulate the lines defined by the optical flow. After it, the largest peak would correspond with the required FoE. In [12] Wu et al. compare different minimization criteria, more specifically, the algebraic method, which is a linear problem, and the geometric method, which is non-linear and numerically more expensive. It is worth noting here that, for the geometric method, the *Cross Ratio*, which is the main tool used in our work, is also employed there, although in a fundamentally different way. In their work, it is employed to generate the so called *inherent constrains* between a pair of points in two consecutive frames. If the inherit constrain fails, that is, the *Cross Ratio* does not hold, the two corresponding pair can not be considered as *true* correspondences, and are eliminated from the FoE computation.

In [13], Bak et al. define the C-Velocity over a planar surface imaged by a camera. When the plane is aligned with the image axis, the optical flow vectors on the plane can be used to estimate the FoE. Although simple, this method can only be used when actual planes are present in the image. For that reason, the use of the algorithm is limited to urban scenes, where planar facades and the road can be used as planes to compute the C-Velocity. In [14], Born projects the optical flow vectors onto the horizontal and vertical axis. These components form a line and the point of intersection with the image axis is the required FoE. A linear regression is needed to compute the line parameters.

Although discrete methods have been usually preferred over continuous ones, these methods also show some disadvantages. On the one hand, finding strong features and correspondences can be a difficult task, and sometimes it could not be present in the scene. Furthermore, these methods are normally less robust because they use local instead of global information.

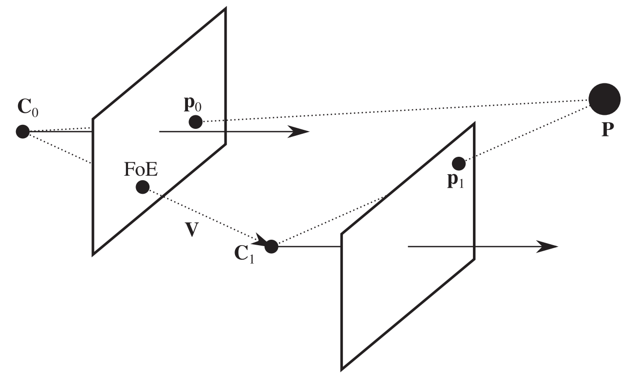

Fig. 1. Focus of expansion of a translating camera. The center of projection is located in $\mathbf{C}_0$ at $t = t_0$, and moves to $\mathbf{C}_1$ at $t = t_1$ with velocity $\mathbf{V} = (V_x, V_y, V_z)^{\mathrm{T}}$. A static point $\mathbf{P} = (P_x, P_y, P_z)^{\mathrm{T}}$ is projected to $\mathbf{p}_0$ and $\mathbf{p}_1$ at t_0 and t_1 respectively. The focus of expansion is the image of the vector $\mathbf{V}$.

An alternative way to compute the FoE is by means of algorithms that compute the 3D camera motion, which are normally referred to as *Visual Odometry* (VO) systems. In this case, the goal is the computation of both the rotational matrix R and the translational vector $\mathbf{v}$, such that the relation between camera positions at two consecutive frames is [15]:

$$\mathrm{T} = \begin{pmatrix} \mathrm{R} & \mathbf{v} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{pmatrix}. \tag{1}$$

When the camera motion is a pure translation, R reduces to the identity matrix, and the image of the vector $\mathbf{v}$ is the FoE, as was shown in Fig. 1. In these systems, for the computation of R and $\mathbf{v}$, typically the *Essential Matrix* E is first computed from a set of correspondences between two consecutive frames. The relationship between those elements is given by:

$$\mathrm{E} = [\mathbf{t}]_{\times}\mathrm{R} \tag{2}$$

where $[\mathbf{t}]_{\times}$ is the matrix representation of the *Cross Product* with $\mathbf{t}$. Thus, the computation of the *Essential Matrix* is a fundamental step in these kind of systems. Many works have been proposed in the VO field. In [16], Forster et al. designed a VO system for Micro Aerial Vehicles (MAVs) using a downward-looking camera. Although it shows accurate results, it is mainly designed for planar surfaces. In the work by Geiger et al. [17], R and $\mathbf{t}$ is computed by iterative minimization, using the Gauss–Newton algorithm, of the projection error of the detected points to the image planes. On top of that procedure, a standard Kalman Filter is used to improve the estimations.

It is worth noting that typically such VO systems are not only designed for motion estimation, but these systems are also able to simultaneously perform a 3D reconstruction of the scenario, a technique called *Simultaneous Localization and Mapping* (SLAM). Nonetheless, we are only interested in the motion estimation block.

As we will see in Section 4, some of these algorithms have been implemented (or downloaded from the paper web page if available), along with the algorithm proposed in this paper, for comparison purposes. For this task, we have employed both, virtual video sequences generated with Blender[1] and a series of challenging video sequences recorded with an on-board camera mounted on a vehicle.

[1] http://www.blender.org/.