HAL
open science

PhD Forum: Camera Pose Estimation Suitable for Smart Cameras

Abiel Aguilar-González, Miguel Arias-Estrada, François Berry

To cite this version:

Abiel Aguilar-González, Miguel Arias-Estrada, François Berry. PhD Forum: Camera Pose Estimation Suitable for Smart Cameras. 11th International Conference on Distributed Smart Cameras, Sep 2017, Stanford, United States. 10.1145/3131885.3131938 . hal-01657273

HAL Id: hal-01657273

https://hal.science/hal-01657273v1

Submitted on 6 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PhD Forum: Camera Pose Estimation Suitable for Smart Cameras

Abiel-Aguilar-González[1,2], Miguel Arias-Estrada[1], François Berry[2]
1. Instituto Nacional de Astrofíosica, Óptica y Electrónica (INAOE),Tonantzintla, Mexico
2. Université Clermont Auvergne (UCA), Institut Pascal, Clermont-Ferrand, France
Contact: abiel@inaoep.mx

ABSTRACT

Camera pose estimation across video sequences is an important issue under several computer vision applications. In previous work, the most popular approach consists on optimization techniques applied over 2D/3D point correspondences for two consecutive frames from a video sequence. Unfortunately, these optimization techniques are iterative and depend on nonlinear optimizations applied over some geometric constraint. For real-time embedded applications, another approach, more efficient in terms of computational size and cost, could be a linear or closed-form solution for the camera pose estimation problem. In this work, we introduce a new approach for camera pose estimation, this approach uses 2D visual features displacements as linear/dependent parameters for the camera pose estimation so, camera pose can be estimated without iterative behavior and without geometric constraints. As result, the proposed algorithm could be implemented inside a small FPGA device, suitable for smart cameras. Preliminary results are encourageous and show the viability of the proposed approach.

Keywords

Camera pose, embedded systems, FPGA

1. INTRODUCTION

In the last few years, camera pose estimation across unknown video sequences appears repeatedly under several computer vision contexts: visual servoing [2], Simultaneous Localization and Mapping (SLAM) [7], augmented reality [5], etc. In previous work, primary interest was for an accurate/generalized solution for the camera pose problem [4, 6]. Unfortunately, these solutions have iterative behavior and requires relatively high computational resources. As result, several SLAM algorithms or augmented reality systems have limitations for embedded applications and, even with the use of relatively powerful PCs, camera pose often is limited between 15-25 estimations per second.

ACM ISBN .
DOI:

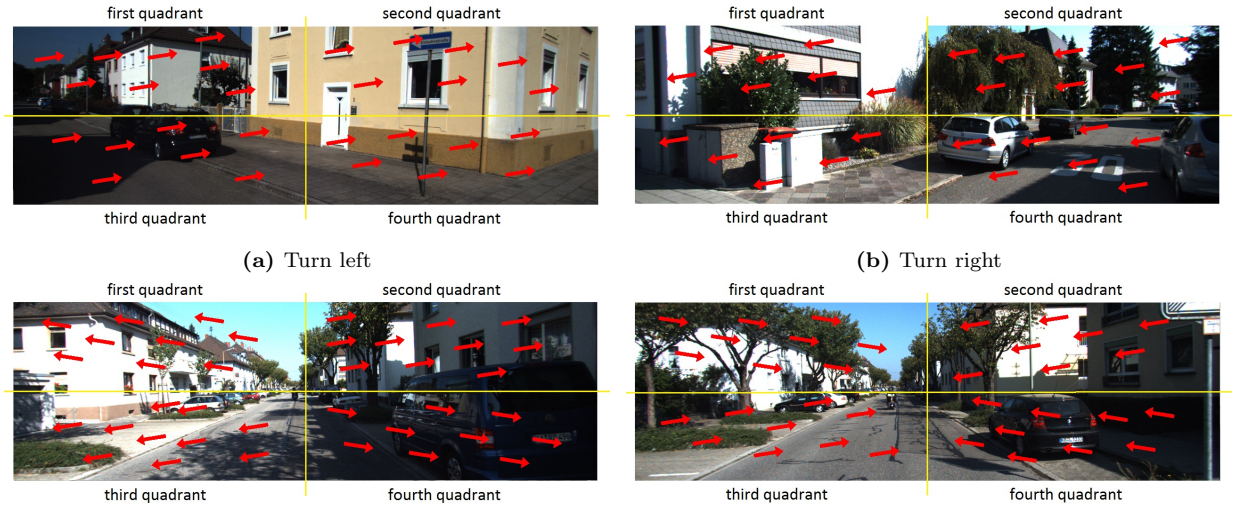
In this work, we believe that a smart camera that delivers camera pose at video cadence could be an important contribution for current embedded computer vision applications. In general, several applications such as robotics, SLAM or augmented reality could reach onboard camera pose processing. Furthermore, given the camera pose algorithm would be inside the camera fabric and close to the image sensor, higher processing speed and lower size/cost could be achieved (compared with previous work).

2. CAMERA POSE ESTIMATION SUITABLE FOR SMART CAMERAS

In smart camera context, one central issue is the implementation of complex/intensive computer vision algorithms inside the camera fabric. For processing purposes, FPGA devices are excellent candidates since they support data parallelism with low power consumption. Furthermore, FPGA devices are smaller and cheaper compared with other devices such as GPU processors or Intel processors. Under this premise, it is possible to assume that the more promising solution for the camera pose estimation is for an FPGA architecture. However, there are several challenges because previous background addressed the camera pose estimation problem via optimization techniques (these are difficult to be implemented inside FPGA fabric). In previous work, some approaches have studied the viability of solutions with small iterative number [10, 9, 8]. Nevertheless, in all cases, mathematical formulation limits its FPGA implementation.

3. CAMERA POSE ESTIMATION BASED ON PARALLEL CLASSIFIERS

As early formulation, we consider an autonomous vehicle scenario in which all possible camera movements are linear dependent of the x, y pixel displacements whiting the scene. i.e., given the input image divided in four quadrants (as shown in **Fig.** 1) and, consider four different cases: turn left, turn right, forward and backward; for the turn left case: all pixels within all the quadrants must have right displacement under the horizontal axe and similar displacement for the vertical axe, see **Fig.** 1a. For the turn right case: all pixels for all the quadrants must have left displacement under the horizontal axe and similar displacement for the vertical axe, as shown in **Fig.** 1b. For the forward case: pixels displacement are to outside the image, see **Fig.** 1c. Finally, for the backward case: pixels displacement are to the image center, as illustrated in **Fig.** 1d.



(a) Turn left **(b)** Turn right

(c) Forward **(d)** Backward

Figure 1: Given an autonomous vehicle scenario. Four different cases: turn left, turn right, forward and backward have to fulfill with all possible movements from a single moving camera installed over the vehicle. In such scenario, pixel displacements inside the images (red arrows) have different directions for each case.

3.1 The proposed approach

Given $p_i\{x, y\}$ spatial locations for i feature points (obtained via FAST, SIFT, Harris, etc.) from frame t of a video sequence, Q that represents the horizontal, vertical displacements for the p_i feature points at frame $t+1$ is computed as shown in **Eq.** 1; where n represents each one of the quadrants from the input image (see **Fig.** 1), g are the feature points number within the n quadrant and k_i are the spatial locations of the p_i feature points at frame $t+1$, these can be obtained via feature tracking algorithms. Considering the Q parameter is linearly/dependent to the camera movement, j known camera displacements (obtained from datasets) can be correlated with its corresponding Q_j parameters, see **Eq.** 2; where $C_j\{x, y, z, \alpha, \theta, \beta\}$ is the camera movement with six degree of freedom. Finally, given each Q_j parameter and its corresponding camera displacement (C_j) as a unique class for a camera movement classifier, and considering that each possible camera movement (taken at some known viewpoints from a known scene) is included in the classifier definition, any camera movement for an unknown viewpoint can be estimated as shown in **Eq.** 3 and 4; where Q' is the Q parameter for the pose being computed and C_k is the camera pose of the class that minimizes the absolute difference between the Q' parameter and the Q_j parameters.

$$Q\{x_n, y_n\} = 1/g \sum_{i=1}^{i=g} (p_i\{x, y\} - k_i\{x, y\}) \tag{1}$$

$$Q_j\{x_n, y_n\} \mathrel{\widehat{=}} C_j\{x, y, z, \alpha, \theta, \beta\} \tag{2}$$

$$k = \arg\min_j(\mathrm{H}), \qquad \mathrm{H}(j) = |Q_j - Q'| \tag{3}$$

$$\mathrm{pose}\{x, y, z, \alpha, \theta, \beta\} = C_k \tag{4}$$

3.2 FPGA architecture

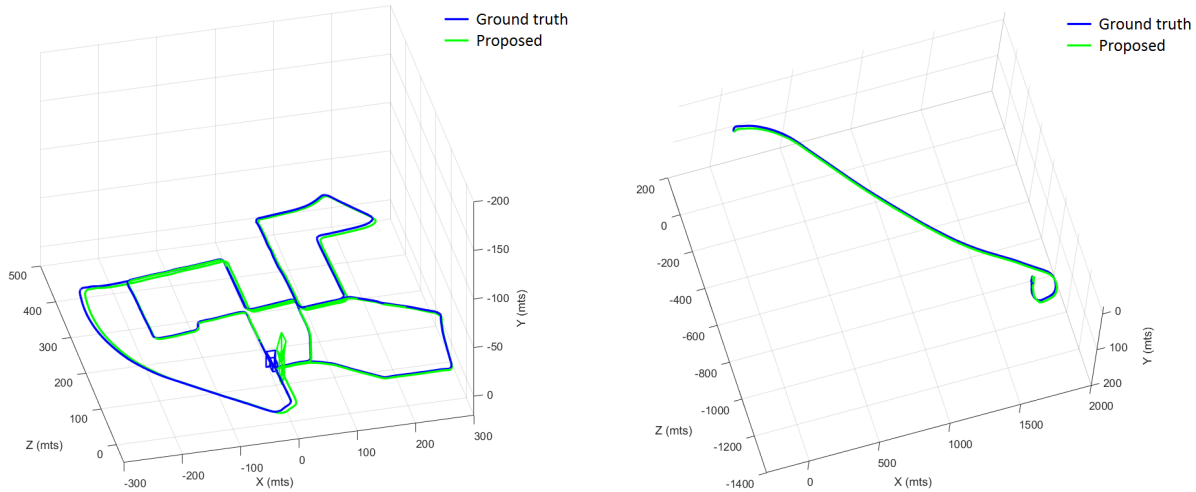
For the FPGA implementation, the proposed approach could be implemented as a multiplexer tree. In this case, definitions for the movement classifier can be saved into logical registers, then, given an input Q' parameter for an unknown camera movement, the multiplexer tree computes the camera position by parallel comparisons between Q' and the stored Q_j parameters.

4. PRELIMINARY RESULTS

We implemented our algorithm in a MatLab R2016b code that uses the KITTI dataset [3] as input sequences. First, feature points (Harris corners) are tracked along the video sequence. Then, several Q_j parameters and its corresponding (C_j) camera displacements are saved. Finally, unknown camera movements can be estimated by comparing the Q parameter of the input frame with previous stored Q_j parameters and (C_j) camera displacements. In **Fig.** 2, results by applying our algorithm over two different KITTI sequences are shown. In both cases 400 random frames from the first video sequence were used in the classifier definition, then, camera pose for the remaining frames in the first sequence (4171 frames) and for all the frames for the second sequence were computed by the proposed algorithm.

5. CONCLUSIONS AND FUTURE WORK

In this work, we have introduced a new camera pose estimation approach in which pixel displacements are used as linear/dependent parameters for the camera pose estimation. Different to previous work, our approach can estimate camera poses without iterative behavior and without geometric constraints. As result, the proposed algorithm could be implemented inside a small FPGA device, suitable for smart cameras. As work in progress, we are generalizing our algorithmic approach and implementing it inside the DreamCam, [1], a robust/flexible smart camera.



(a) **(b)**

Figure 2: Camera pose estimation for the first two video sequences of the KITTI dataset [3]. In both cases, the proposed algorithm achieves accurate results (similar with the ground truth).

6. REFERENCES

[1] M. Birem and F. Berry. Dreamcam: A modular fpga-based smart camera architecture. *Journal of Systems Architecture*, 60(6):519–527, 2014.

[2] F. Chaumette, S. Hutchinson, and P. Corke. Visual servoing. In *Springer Handbook of Robotics*, pages 841–866. Springer, 2016.

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[4] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[5] I. Hong, G. Kim, Y. Kim, D. Kim, B.-G. Nam, and H.-J. Yoo. A 27 mw reconfigurable marker-less logarithmic camera pose estimation engine for mobile augmented reality processor. *IEEE Journal of Solid-State Circuits*, 50(11):2513–2523, 2015.

[6] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[8] Y. Nakajima and H. Saito. Robust camera pose estimation by viewpoint classification using deep learning. *Computational Visual Media*, 3(2):189–198, 2017.

[9] A. Rubio Romano. 3d pose estimation using convolutional neural networks. Master's thesis, Universitat Politècnica de Catalunya, 2015.

[10] F. Shi and Y. Liu. Estimation of camera pose using 2d to 3d corner correspondence. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 2, pages 805–809. IEEE, 2004.