Accurate scale estimation based on unsynchronized camera network

Rawia Mhiri, Pascal Vasseur, Stephane Mousset, Rémi Boutteau, Abdelaziz Bensrhair

ACCURATE SCALE ESTIMATION BASED ON UNSYNCHRONIZED CAMERA NETWORK

Rawia Mhiri, Pascal Vasseur, Stéphane Mousset, Rémi Boutteau, Abdelaziz Bensrhair

LITIS laboratory, INSA de ROUEN - Université de Rouen
IRSEEM, ESIGELEC

ABSTRACT

In this paper we present an unsynchronized camera network able to estimate the motion and the structure with accurate absolute scale. The proposed algorithm requires at least three frames: two frames from one camera and a frame from a neighbouring camera. The relative camera poses are estimated with classical Structure-from-Motion and the absolute scales between views are computed by assuming straight trajectories between consecutive views of one camera. We propose a final optimisation step to refine only the scale and the 3D points. Our method is evaluated in real conditions on the KITTI dataset. We show quantitative evaluation through comparisons against GPS/INS ground truth.

Index Terms— motion estimation, scale estimation, Structure-from-Motion, scale estimation, ego-motion

1. INTRODUCTION

Long sequences occur to be an important challenge for motion estimation applications. Indeed small errors issues from the estimation process are accumulated over time, which cause a drift in the estimated trajectories. To handle this problem, a bundle adjustment (BA) can be applied either globally for all the sequence, known as global bundle adjustment, or locally for some viewpoints, which is called windowed bundle adjustment. Monocular systems require an initialisation step or a prior knowledge to obtain the absolute metric scale. Multi-camera systems can be a good alternative for scale estimation while also allowing to cover the whole surrounding area around a vehicle. However, if epipolar geometry is required for some algorithms like in [1], the necessary synchronization device becomes a major inconvenience. The synchronization device is difficult to embed in practical. Asynchronous camera network presents several advantages. First, it can be developed with low cost devices and close-to-market sensors, which is often suggested for real applications [2]. Next, the acquisition does not depend on the slowest camera or the synchronisation device itself and images can be acquired continuously from each camera. The bandwidth constraint that appears in the case of synchronized cameras is also avoided and finally the network can be easily modified.

Our previous work introduced a new method relaxing the synchronization constraint that we called the "triangle-based" method [3]. This method was based upon the assumption that the motion between two consecutive frames is rectilinear. Although "triangle-based" method shows good performance for a whole sequence, notable errors occur in presence of curves. These errors, which are caused by the straight motion hypothesis, reduce the accuracy of the absolute scale estimation. If the scale estimation is not very well performed, larger error will be introduced to the motion and the structure computation. To improve the scale estimation part, we suggest a final absolute scale optimization step and thus provide a more complete and accurate unsynchronized camera network. Our main contribution is that we reduce errors imposed by the straight motion assumption of the triangle-based method. To focus on the scale estimation refinement, we suppose that the initial estimated rotation matrices and the translation vectors obtained with the 5-points algorithm [4] are fixed and we only optimize the scale factors and the 3D structure.

The rest of the paper is organized as bellow: in section 2, we present a short discussion about the previous work related to motion estimation and optimisation process. Next, we give a detailed description of the triangle method and present bundle adjustment applied on our camera network in section 3. Before concluding our work in the last section, experiments and results for real sequences are presented in section 4.

2. RELATED WORK

Motion estimation and structure from motion algorithms have been extensively studied [1] [5] [2] [6]. 2D-2D pose estimation can be obtained with the classical 5-points algorithm [4]. An initial estimation from keypoints always suffers from errors accumulated over time, and leads to drifts that even robust algorithms can not avoid. To handle this problem, optimization process as BA is usually applied as the last step of Structure-from-Motion applications. In visual odometry, a loop closure detection followed by a global bundle adjustment is usually used [7]. This technique is very efficient but difficult to apply in real applications. However, it is possible to use a local bundle adjustment with a limited temporary window [8].

Asynchronous camera systems are rarely studied for the

motion estimation problem. In [9], an unsynchronized camera system was presented a structure from stereo vision for SLAM. This method used three images : two from the left camera at the first and the third time steps and one from the right camera in the time lap between the left images. Features from the three images are interpolated to create the missing synchronized image assuming that the features change linearly between the frames. Using the robot poses from odometers, this method allows to have an accurate 3D structure but it does not estimate the scale or the motion parameters.

If the absolute metric scale is desired, visual odometry process must integrate a particular 3D knowledge. For monocular systems, a prior 3D knowledge is required as initialization and to be maintained later. In [6], Fraundorfer et al. present a constricted BA parametrization for relative scale estimation. To compute the rotation and the translation parts, authors use the 1-point algorithm [10] based on the general Ackermann steering principle for circular motion. To compute the scale, Fraundorfer et al. propose a global parametrization process to solve for all the scales at once. Instead of optimising all the motion parameters like the classical BA algorithm, they refine only the relative scales. The main difference between [6] and our work is that they use a monocular camera system and we developed an asynchronous multi-camera system. Also, their algorithm is based on circular motion assumption and our method makes a straight motion assumption instead.

3. MOTION AND STRUCTURE ESTIMATION

The proposal method is divided in two main steps : the motion and the structure estimation with the absolute metric scale and the refinement of the scale and the structure by BA.

3.1. TRIANGLE-BASED METHOD

In our previous work [3], the approach is based on two assumptions : two consecutive frames of each camera follow a linear trajectory and the neighbouring cameras have a common filed of view. Our method can be generalized to N cameras and requires at least two cameras. In the rest of the paper, we introduce approach for two cameras C_i and C_j. The relative cameras poses are computed via Structure-from-Motion and the absolute scale factors are computed using the extrinsic calibration and the linearity assumption.

Three relative poses : rotation matrices R and translation unit vectors t, are computed between the three images using the 5-points algorithm. Figure 1(a) shows the triangle shape between the camera C_i in the time step 0 and 2, and the camera C_j in the time step 1. Red lines refer to the transformations between the cameras position and the green line refers to the calibration process transformation. T_{i2}^{i0} is the transformation of the camera coordinate system of C_{i2} to the camera coordinate system of C_{i0}, the same for T_{j1}^{i0} and T_{j1}^{i2}. The transformation T_{i1}^{j1} allows a static coupling of the two cameras in the same time because of the system rigidity.

(a) The big triangle (b) The sub-triangles

Fig. 1. Triangle-based method for unsynchronized cameras, blue cameras are cameras which take images: C_{i0}, C_{j1}, and C_{i2}. C_{i1} is a virtual position of the camera C_i.

We integrate T_{i1}^{j1} in the triangle shape to deduce the absolute scale factors. The virtual pose of the camera C_i at the time step 1, C_{i1}, is intercalated in the big triangle. This pose gives two sub-triangles shapes, figure 1 (b). In our three triangles shape, we can write these equations:

$$\begin{cases} T_{i1}^{i0} = T_{j1}^{i0} T_{i1}^{j1} \Longrightarrow \begin{bmatrix} R_{i1}^{i0} & \lambda_1 t_{i1}^{i0} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{j1}^{i0} & \alpha t_{j1}^{i0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{i1}^{j1} & t_{i1}^{j1} \\ 0 & 1 \end{bmatrix} \\ T_{i1}^{i2} = T_{j1}^{i2} T_{i1}^{j1} \Longrightarrow \begin{bmatrix} R_{i1}^{i2} & \lambda_2 t_{i1}^{i2} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{j1}^{i2} & \beta t_{j1}^{i2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{i1}^{j1} & t_{i1}^{j1} \\ 0 & 1 \end{bmatrix} \\ T_{i2}^{i0} = T_{j1}^{i0} T_{i2}^{j1} \Longrightarrow \begin{bmatrix} R_{i2}^{i0} & (\lambda_1+\lambda_2) t_{i2}^{i0} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{j1}^{i0} & \alpha t_{j1}^{i0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{i2}^{j1} & \beta t_{i2}^{j1} \\ 0 & 1 \end{bmatrix} \end{cases} \quad (1)$$

Expanding this equations and decoupling the rotation and translation terms, we obtain the equation 2:

$$\begin{bmatrix} t_{i1}^{i0} & 0 & -t_{j1}^{i0} & 0 \\ 0 & t_{i1}^{i2} & 0 & -t_{j1}^{i2} \\ t_{i2}^{i0} & t_{i2}^{i0} & -t_{j1}^{i0} & -R_{j1}^{i0} t_{i2}^{j1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} R_{j1}^{i0} t_{i1}^{j1} \\ R_{j1}^{i2} t_{i1}^{j1} \\ 0 \end{bmatrix} \quad (2)$$

The absolute scales factors, λ_1, λ_2, α, and β can be derived by means of a linear least square model. The linear system can be written as $A.X = B$ where X is the vector of the scale factors. The translation of the camera position C_{i0} to the coordinate system of C_{i1} and the translation of C_{i2} to C_{i1} are obtained thanks to the linearity assumption. For more details about the triangle based method please refer to [3].

3.2. BUNDLE ADJUSTMENT

BA has an important role in computer vision applications focused on 3D reconstruction and Structure-from-Motion, in that it allows to refine the parameters of the motion and the 3D structure describing the environment [11]. This is an optimisation of both the 3D points positions and the camera parameters, so the reprojection errors between observed image-points and projected points from the 3D structure are minimized. In our approach, we apply BA for some views locally. To focus on the scale estimation refinement, we suppose that the initial estimated rotation matrices and the translation vectors are fixed and we only optimize the scale factors and the 3D structure. We choose to take a sliding window composed by two consecutive triangles (five images). We compute the 3D

points at scale from the inliers corresponding points and the camera poses expressed on the first camera of each window.

3.2.1. Problem formulation

In the pinhole camera model, the projection function of a 3D point X of a scene into a 2D point x in the image plane can be written using a perspective transformation like $x = K[R|s*t]X$ where, K is the matrix of intrinsic parameters, $[R|s*t]$ is the matrix of extrinsic parameters, s design the scale factor, and t the unit translation vector.

BA minimizes the reprojection errors between observed image-points and projected points from the 3D structure. The minimization of the reprojection error is resolved using non-linear least-squares algorithms such as LevenbergMarquardt algorithm [12]. This algorithm requires to obtain successive approximations of the parameters vector P according to the algorithm of LevenbergMarquardt 1, where $\triangle_i$ is obtained by solving the augmented normal equation for each iteration. The reprojection errors are computed and evaluated for both P_i and P_{i+1}.

Algorithm 1 Pseudo-code of LevenbergMarquardt algorithm

```
i←0
λ←0.001
compute ‖e(P_0)‖
while i < MAX_ITERATIONS and ‖e(P_i)‖> threshold do
    solve the augmented normal equation:
    (J^T J+λI)△=−J^T ε
    evaluate the new parameters vector P_{i+1}=P_i+△_i
    if ‖e(P_{i+1})‖≥‖e(P_i)‖ then
        λ←10λ
    end if
    if ‖e(P_{i+1})‖≤‖e(P_i)‖ then
        λ←λ/10, P_i=P_{i+1}
    end if
    i=i+1
end while
```

3.2.2. Jacobian Derivation

In our proposed method, the Jacobian matrix J is calculated by derivating the projection function only by the scale factor s and the 3D point X. The symbolic differentiation of the projection function F is performed with respect to the scale factor s and to the 3D coordinates of the 3D point X. The Jacobian matrix is the derivative of the projection matrix by the 3D point X and the scale factor s. Differentiating the projection function by the 3D point, we obtain the derivatives JX (2x3 matrix) with respect to X, the equation 3. When we derive the projection function by the scale factor, we obtain JS (2x3 matrix).

$$JX = [\frac{\partial F}{\partial X}] \text{ and } JS = [\frac{\partial F}{\partial s}]. \quad (3)$$

For each 3D point, we calculate its JX and its JS for all the cameras included into the sliding window. J will have a sparse block structure, Figure 2. If we consider n cameras and m 3D points, the jacobian will be a $2*n*m$x$1*n+3*m$ matrix.

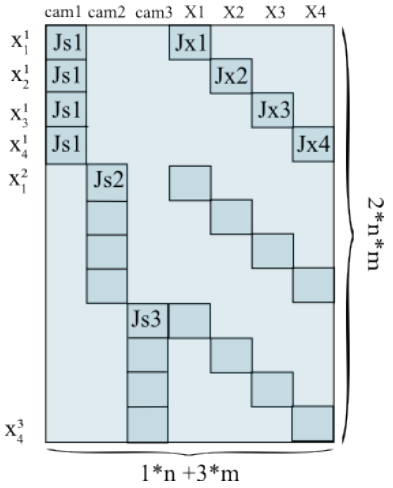


Fig. 2. Jacobian matrix for a bundle adjustment problem consisting of 3 cameras and 4 points. the dark blue means the camera parameters JS and the 3D point parameters JX, the light blue means the zeros elements

4. EXPERIMENTS

In this section, we present the results of our algorithm on a real world image sequence from the KITTI dataset [13] [14]. The sensors used in our comparison are the gray scale synchronized stereo cameras and the GPS/IMU inertial navigation system. To have the unsynchronized fact, we use only one image in every time step. For each set of three images, key-points are extracted with SURF detector [15] and described with Freak descriptor [16]. Each corresponding points between two images allows to estimate an essential matrix using the 5-point algorithm implementation. The relative rotations and translations are recovered using cheirality check. The retained inliers are then triangulated to compute the 3D points. We solve the equation system of equation 2 to compute the initial absolute scale factors λ_1, λ_2, α, and β and obtain the camera poses at scale. For two consecutive triangles (five images), we compute the 3D points at scale from the inliers corresponding points and the camera poses expressed on the first camera of the sliding window. Afterwards, we apply the Levenberg Marquard algorithm. Our results are presented for some images of the sequence 0 of the KITTI dataset.

4.1. Results of an "optimal" BA

For the quantitative evaluation of our proposal algorithm, we apply the BA on the ground truth (GT) poses. For a sliding window of two triangles, we take the rotation matrix, the

translation unit vector and the scale factors of the camera from the GPS GT. Then we compute 3D points at scale and apply the BA step. This test, which we call the "optimal" BA, gives the reference results for the scale factors refinement after the BA step. To evaluate the scale factors, we compute the ratio of the evaluated scale factor by the scale factor of the GT, Both before and after BA as in equations 4 and 5. we obtain almost the same starting data with acceptable small errors. We add a gaussian noise to the scales values to evaluate our method. the trajectory obtained when accumulating the poses after BA is almost the same of the GT trajectory. The obtained results are summarized on the table 1 where scale 1 is the scale factor of the transformation of the second to the first camera of the window, scale 2 is the scale of the transformation of the third camera to the first one, same for scale 3 and scale 4.

$$\text{ratios before} = \frac{\text{evaluated scale factors before BA}}{\text{scale factors GT}} \tag{4}$$

$$\text{ratios after} = \frac{\text{evaluated scale factors after BA}}{\text{scale factors GT after optimal BA}} \tag{5}$$

Table 1. Ratios before and after a BA step for gaussian added noise ($\sigma = 0.01$)

	scale 1	scale 2	scale 3	scale 4
Before BA	1.0014	0.9993	0.9996	0.9994
After BA	1.0003	0.999803	01.0001	1.003

The ratios are around 1 so we judge that our BA algorithm gives very accurate results. These errors are due to many reasons : inaccuracy of detector and descriptor algorithms, errors of matching, computation errors of the triangulation and the reprojection of the 3D points into the 2D image points. The BA step minimizes reprojection errors to improve the scale factors and the 3D points. When the motion parameters are "perfect" (GT), it remains some errors on the structure.

4.2. Results of Triangle based method

We apply the proposed BA on the Triangle based method. Figure 4 shows the distribution of the accumulated reprojection errors before and after the BA for a sequence of 200 images.We compute the ratio same as equations 4 and 5. Results are summarized in table 2. the trajectories are presented in figure 5. The figure shows that the trajectory of the proposal algorithm is better than the trajectory obtained in our previous work.

5. CONCLUSION

In this paper, we presented a complete motion estimation method which we called "triangle-based" method using an unsynchronized multi camera setup. The "triangle-based" method motion estimation presented in our previous work [3] assumes that the trajectory between two consecutive frames of one camera is rectilinear and requires an off line calibration knowledge. The linearity assumption causes small errors especially for curve trajectories. The presented approach results improves the accuracy on the absolute scale estimation.

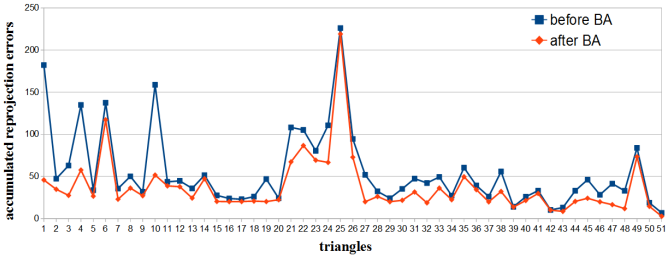
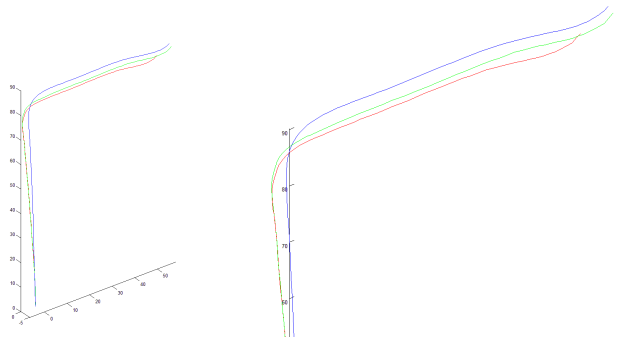


Fig. 3. Accumulated reprojection errors for 52 triangles for a BA applied on the triangle based method estimation

Fig. 4. Example of reprojection errors of the 3D points after BA, each color refers to the reprojection errors on a camera of the sliding window (5 cameras)

Table 2. Ratios before and after a BA step for triangle based method

	scale 1	scale 2	scale 3	scale 4
Before BA	0.9516	0.944478	0.9555	0.9500
After BA	1.037	1.002	1.0286	1.0609



(a) trajectory of 200 images (a) zoom in on the trajectory

Fig. 5. trajectory of 200 images for two triangles : the estimation in red, GT in blue, BA in green

6. REFERENCES

[1] T.-W. Hui, "Structure from motion directly from a sequence of binocular images without explicit correspondence establishment," in *Image Processing (ICIP), 2013 20th IEEE International Conference on*. IEEE, 2013, pp. 3607 – 3611.

[2] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Muhlfellner, S. Wonneberger, J. Timpner, S. Rottmann, B. Li *et al.*, "Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 809–816.

[3] R. Mhiri, P. Vasseur, S. Mousset, R. Boutteau, and A. Bensrhair, "Visual odometry with unsynchronized multi-cameras setup for intelligent vehicle application," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 1339–1344.

[4] D. Nistér, "An efficient solution to the five-point relative pose problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 756–770, 2004.

[5] G. H. Lee, F. Faundorfer, and M. Pollefeys, "Motion estimation for self-driving cars with a generalized camera," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 2746–2753.

[6] F. Fraundorfer, D. Scaramuzza, and M. Pollefeys, "A constricted bundle adjustment parameterization for relative scale estimation in visual odometry," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1899–1904.

[7] M. Lhuillier, "Automatic scene structure and camera motion using a catadioptric system," *Computer Vision and Image Understanding*, vol. 109, no. 2, pp. 186–203, 2008.

[8] E. O.-B. Alfredo Ramirez and M. Trivedi, "Panoramic stitching for driver assistance and applications to motion saliency-based risk analysis," in *Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013.

[9] M. Svedman, L. Goncalves, N. Karlsson, M. Munich, and P. Pirjanian, "Structure from stereo vision using unsynchronized cameras for simultaneous localization and mapping," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 3069–3074.

[10] D. Scaramuzza, F. Fraundorfer, and R. 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.

[11] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustmenta modern synthesis," in *Vision algorithms: theory and practice*. Springer, 2000, pp. 298–372.

[12] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of applied mathematics*, vol. 2, pp. 164–168, 1944.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and PatternRecognition (CVPR)*, 2012.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[16] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. Ieee, 2012, pp. 510–517.