Visual odometry with unsynchronized multi-cameras setup for intelligent vehicle application

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

Visual odometry with unsynchronized multi-cameras setup for intelligent vehicle application

Rawia Mhiri[1], Pascal Vasseur[1], Stephane Mousset[1], Remi Boutteau[2] and Abdelaziz Bensrhair[1]

***Abstract*— This paper presents a visual odometry with metric scale estimation of a multi-camera system in challenging unsynchronized setup. The intended application is in the field of intelligent vehicles. We propose a new algorithm named "triangle-based" method. The proposed algorithm employs the information from both extrinsic and intrinsic parameters of calibrated cameras. We assume that the trajectory between two consecutive frames of a camera is a linear segment (straight trajectory). The relative camera poses are estimated via classical Structure-from-Motion. Then, the scale factors are computed by imposing the known extrinsic parameters and the linearity assumption. We verify the validity of our method both in simulated and real conditions. For the real world, the motion trajectory estimated for image sequence of two cameras from KITTI dataset is compared against the GPS/INS ground truth.**

I. INTRODUCTION

Driving Assistance systems (DAS) and parking assistance systems require a perfect knowledge of the vehicle vicinity in order to be efficient and safe. Vision sensors allow to obtain such information and different configurations can be used such as mono-camera [2][7], stereovision [4][16] and multi-camera systems [1][3][13]. These systems allow then to perform applications such as obstacle and road detection, localization and mapping. However, a common very important step in these applications deals with the motion estimation also known as visual odometry [5].

The cameras poses computation forms the basis of visual odometry. It requires the computation of the rotation, translation and relative scale between consecutive frames. The estimation of the scale is the most sensitive part and needs to be robustified. If the absolute metric scale is desired, the visual odometry process must then integrate a particular 3D knowledge. In the case of a monocular system, a prior knowledge about the 3D scene is necessary as initialization and then maintained during the motion by a constricted parametrization of the bundle adjustment step for example [8]. When at least two cameras are used, the epipolar geometry constraint can then be used in order to deduce the metric scale [17]. Nevertheless, it is only possible if whole set of cameras is synchronized and consequently that the scene is captured at the same time from the different points of view.

However, in automotive applications based on low cost devices and close-to-market sensors such the one developed in [4], the synchronization is a major inconvenience. It uses an additional electronic circuitry and expensive hardware, therefore making it unsuitable. Additionally, asynchronous acquisition can be easily implemented without depending on the slowest camera for example and also simplifies the problem of bandwidth for the transmission of the images. Up to now, unsynchronized cameras have been rarely studied [10][15]. For unsynchronized cameras and moving system, the motion estimation suffers from many problems among which accumulated error due to inaccurate scale estimation is the most sensitive one.

To solve this problem, we introduce a new method relaxing the synchronization constraint that we called the "triangle-based" method. Our main contribution is the estimation of the motion with an accurate metric scale using an asynchronous multi camera ring without any temporal calibration. This method is based upon the hypothesis that the motion between two consecutive frames is rectilinear (straight or smoothly curved). This assumption is a good approximation of a smooth trajectory between consecutive positions. We also consider that the complete geometric calibration process is done off line.

The remainder of this paper is organized as follows. After a short discussion about the previous work related on motion and metric scale estimation, we present an overview of our system. Next, we describe precisely our method for the metric motion. Before concluding our work in the last section, experiments and results are presented for real and synthetic sequences.

II. RELATED WORK

Our work aims at solving the motion estimation problem for an asynchronous multi-camera system. Estimating the relative motion of two calibrated cameras is a classical problem. A basic method for such problem is the classical 5-point algorithmic solution proposed by D. Nister [5]. This algorithm finds all the possible solutions for relative camera poses between two calibrated cameras using 5 points correspondences. In [6], Li and Hartley simplify the five point motion estimation algorithm based on the hidden variable resultant technique.

Various configurations with multiple cameras have been studied in order to estimate the motion and to derive the metric scale. To overcome the problems of monocular setup, the classical stereo pair is widely used with a large field of view that enhances the computational accuracy of the relative poses and improves the robustness. For a setup using close-to-market multi-cameras, Lee et al. estimate the motion for self driving cars with a generalized camera model relying

[1] LITIS laboratory rawia.mhiri@insa-rouen.fr
pascal.vasseur@univ-rouen.fr
stephane.mousset@insa-rouen.fr
abdelaziz.bensrhair@insa-rouen.fr
[2] IRSEEM remi.boutteau@esigelec.fr

on 4 cameras [1]. The relative motion is obtained using this generalized camera from a set of synchronized images.

In [9], the authors relax the constraint of overlapping field of views. A 6 Dof stereo visual odometry and structure estimation of a synchronous multi-camera system in an indoor environment is obtained from two cameras. Monocular motion estimation up-to-scale is done in the two cameras, and a linear solution is used to recover the metric scale by imposing the known static transformation between both sensors.

All previous works are able to estimate the movement with scale. Nevertheless, every multi-camera stereo setup uses synchronized cameras. Unsynchronized cameras can be also used to perform such tasks. In fact, there are very few envisaged and possible solutions. It is mainly due to the fact that the unsynchronized setup requires additional informations to perform the motion estimation task. One possible solution can be obtained by 3D-3D approach : each camera perform monocular odometry then a 3D-3D correspondence can be done. The problem here is that the relative scales are different. This solution is not very appropriated for real time applications. Another solution using 3D-2D approach is possible only if a known 3D structure is available. This solution can be used in cartography and mapping but not for a vehicle in an unknown environment.

An unsynchronized multi camera setup was used in structure from Stereo vision for SLAM in [10]. In an indoor environment, the robot makes the 3D structure reconstruction using two unsynchronized cameras. Three images are used to create the 3D structure : two images from the left camera at the first and the third time steps and one image from the right camera in the time lap between the two other images. Virtual synchronized image is created for the left camera in the time lap between the two other images by interpolating feature points from the two left images and the right image in order to use geometric and probabilistic criteria. The method assumes that the features change linearly between the left frames. To obtain the pose difference between the images, this approach uses wheel odometers. Although this method allows a 3D structure to Visual SLAM, it does not estimate the motion of the robot.

Our approach focuses on metric motion estimation with unsynchronized and calibrated cameras. The approach is based on two assumptions. The first assumption is that two consecutive frames of each camera follow a linear trajectory. The trajectory is therefore approximated to a succession of straight segments. In addition, it is assumed that two neighbouring cameras have a common field of view and see the same scene.

III. SYSTEM OVERVIEW

Our system is inspired from the multi camera setup designed by Meilland in [3]: the scene can be seen by two or more cameras. Every two neighbouring cameras share a common field of view in order to have a full view all around the vehicle. The proposed model is flexible, the number of cameras can be changed conveniently. The setup

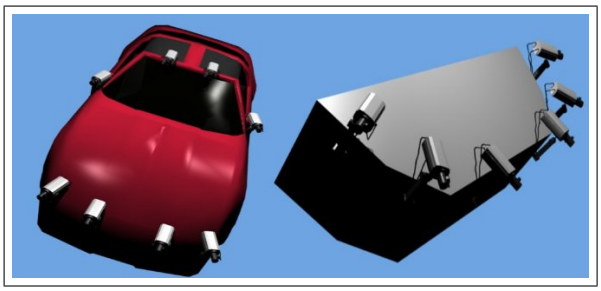

Fig. 1. Example of a multi camera setup

can be mounted on any mobile platform depending on the application. Fig. 1 illustrates an example of a multi camera setup.

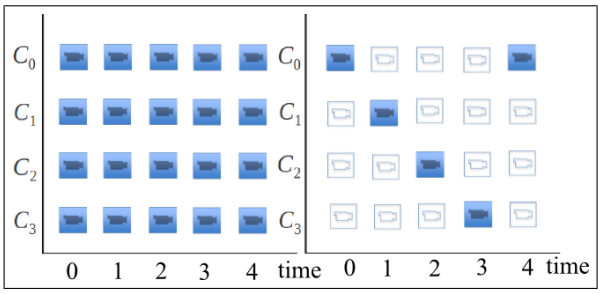

Fig. 2. (a) synchronized N cameras (b) unsynchronized N cameras in the ideal cases

The system is composed by N unsynchronized cameras, i.e. every camera acquires an image independently from the other cameras. The Fig. 2 shows a simple distribution of a synchronized camera system (i.e. all cameras acquire images in the same time) and an unsynchronized one (i.e. only one camera acquires a frame in every time step. The trajectory between two consecutive frames of the same camera is very close to a linear segment line.

IV. MOTION AND METRIC SCALE ESTIMATION : "TRIANGLE-BASED" METHOD

We separate the method into two parts : relative motion estimation and the absolute scale estimation. The relative camera poses are estimated via Structure-from-Motion. Next, we calculate the absolute scale factors using the extrinsic calibration and the linearity assumption.

The cameras acquire the images at different time steps during the displacement of the system. In a series of time steps, the system acquires a sequence of images from different cameras. To simplify the problem, we consider a system with only two cameras which acquire images in three time steps.

The first camera and the second cameras, C_i, C_j respectively acquire the images I_i and I_j. The time notion is placed like an index attached to the camera notation. For example, the camera position at the first time step 0 is denoted by C_{i0}, at the time step 1 by C_{i1}.

The transformations between the camera poses are denoted by T that expresses the rotation 3x3 matrix (denoted by R) and the unit translation 3x1 vector (denoted by t). The

euclidean transformation matrix T is the position of one camera with respect to the coordinate frame of one other camera. For example, the transformation of the first camera position at the time step 0, C_{i0}, to the second camera position at the time step 1, C_{j1}, is T_{i0}^{j1}. Similar to this notation, the rotation matrix and the unit translation vector of the second camera position at the time step 1 , C_{j1}, are expressed in the system coordinate of the first camera at the time step 0, C_{i0}, by R_{j1}^{i0} and t_{j1}^{i0}.

A. Pose estimation

We start with the camera pose estimation. Two relative poses of a camera can be described by the unit translation vector t, the rotation matrix R and the scale factor λ . The transformation between two positions can be written as below:

$$T = \begin{bmatrix} R & \lambda t \\ 0 & 1 \end{bmatrix} \tag{1}$$

The rotation R and the translation t can be obtained by decomposing the essential matrix between two images written as in eq. (2).

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

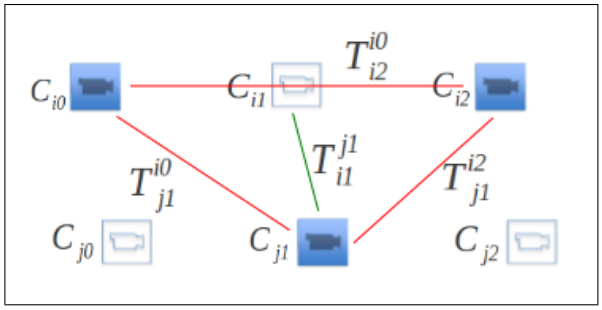

Fig. 3. "Triangle-based" method for unsynchronized cameras : red lines refer to the transformations between the cameras position and the green line refers to the calibration process transformation

Our "triangle-based" method relies on the features points matching in order to estimate the essential matrix. Three essential matrices are computed between three frames : two consecutive frames from the same camera (the first camera) and one frame from the other camera (the second camera) as is modeled in Fig. 3. The Fig. 3 shows all possible transformations between the three frames of the "triangle". Three transformations are determined via SFM: T_{i2}^{i0}, T_{j1}^{i0}, and T_{j1}^{i2}. The lines connecting the real cameras in Fig. 3 refer to those transformations between the cameras position C_{i0}, C_{j1} and C_{i2}. The static transformation T_{i1}^{j1} of the second camera to the first camera can be obtained from the extrinsic calibration process. The line connecting the camera C_{j1} and the virtual camera C_{i1} in Fig. 3 refers to the calibration process transformation.

To determine the camera poses, we start our algorithm by extracting and matching interest points between the frames of the "triangle". In our implementation, we use the SURF detector [11] and the FREAK descriptor [12].

There exists many algorithms which allow to obtain the transformation between the different poses. For calibrated cameras, the 5-point algorithm [5],[6] can be used. To find a linear solution, the 8-point algorithm or its variance can be used [17]. This algorithm allows to compute the fundamental matrix and finds inliers between two frames. Then, the essential matrix can be computed from the fundamental matrix as in eq. (3).

$$E = K'^{\top} F K \tag{3}$$

Where K' and K are the intrinsic calibration matrices of the two cameras. The essential matrix is decomposed by the singular value decomposition to determine the rotation matrix and the translation vector between the two camera's coordinate system. These steps are applied between the positions C_{i0}, C_{i2}, and C_{j1} in order to calculate the transformations T_{i2}^{i0} between C_{i0} and C_{i2}, T_{j1}^{i0} between C_{j1} and C_{i0}, and T_{j1}^{i2} between C_{j1} and C_{i2}.

While the vehicle moves, there is an intermediate position of the first camera at C_{i1}, when the second camera captures an image. However, first camera does not take an image at C_{i1} because these two cameras are unsynchronized. Because of the system is rigid, the transformation T_{i1}^{j1} between these two cameras, C_{i1} and C_{j1}, can be obtained from the extrinsic calibration. As camera C_{i1} itself does not exist in reality, we treat it as a virtual camera. So that, we can estimate the C_{i1}'s position to strongly make our metric scale inference.

To summarize, we will use four transformations : three calculated via SFM (T_{i2}^{i0}, T_{j1}^{i0}, and T_{j1}^{i2}) and a static one (T_{i1}^{j1}) which is assumed to be known from the calibration process.

B. Scale estimation

In the unsynchronized case, the scale factor is unknown between the camera poses. The first assumption expresses the static coupling of the two cameras in the same time step. The second assumption links two positions of the same camera in two time steps. This assumptions allow us to modulate our system by four main transformations as it has been shown in Fig. 3. The three frames (I_{i0}, I_{i2}, and I_{j1}) make a "big triangle" shape between the positions of C_{i0}, C_{i2}, and C_{j1}.

The virtual pose of the camera C_i at the time step 1 (C_{i1}) can be intercalated in the "big triangle". This pose gives two triangular shapes. There are two "sub-triangles" : the first is between C_{i0}, C_{i1}, and C_{j1} and the second one is between C_{i1}, C_{i2}, and C_{j1}.

In the first "sub-triangle", Fig. 4, we can write the transformations as in the eq. (4). the transformation of C_{i1} to the coordinate system of C_{i0}, T_{i1}^{i0} is equal to the transformation of the camera position C_{j1} to the coordinate system of C_{i0}, T_{j1}^{i0}, transformed via the transformation of camera position C_{i1} to the coordinate system of C_{j1}, T_{i1}^{j1}.

$$T_{i1}^{i0} = T_{j1}^{i0} T_{i1}^{j1} \tag{4}$$

The euclidean transformations are expressed at the homogeneous coordinate system as shown in eq. (1). The static transformation T_{i1}^{j1} obtained from the extrinsic calibration is

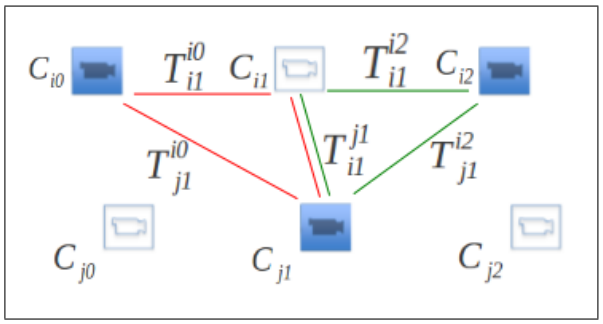

Fig. 4. The first "sub-triangle" : C_{i0}, C_{i1}, and C_{j1} and the second "sub-triangle" : C_{i1}, C_{i2}, and C_{j1}

known with the scale. Expanding the eq. (4), we introduce the unknown scales factors (λ_1, α) of each transformation in eq. (5) : λ_1 is the associated scale factor to the transformation T^{i0}_{i1} and α is the associated scale factor to the transformation T^{j1}_{i0} .

$$\begin{bmatrix} R^{i0}_{i1} & \lambda_1 t^{i0}_{i1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^{i0}_{j1} & \alpha t^{i0}_{j1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^{j1}_{i1} & t^{j1}_{i1} \\ 0 & 1 \end{bmatrix} \tag{5}$$

In eq. (5), we decouple the rotation and the translation terms after expanding. The obtained equations are as follows:

$$R^{i0}_{i1} = R^{i0}_{j1} R^{j1}_{i1} \tag{6}$$

$$\lambda_1 t^{i0}_{i1} - \alpha t^{i0}_{j1} = R^{i0}_{j1} t^{j1}_{i1} \tag{7}$$

Furthermore, the equation eq. (7) can be rewritten as :

$$\begin{bmatrix} t^{i0}_{i1} & -t^{i0}_{j1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \alpha \end{bmatrix} = R^{i0}_{j1} t^{j1}_{i1} \tag{8}$$

The Fig. 4 illustrates the second "sub-triangle" : C_{i1}, C_{i2}, and C_{j1} . Now, we can write the transformations as in the eq. (9). The transformation of the camera's position C_{i1} to coordinate system of C_{i2}, T^{i2}_{i1}, is equal to the transformation of the camera's position C_{j1} to coordinate system of C_{i2}, T^{i2}_{j1}, transformed via the static transformation between the two cameras, T^{j1}_{i1}.

$$T^{i2}_{i1} = T^{i2}_{j1} T^{j1}_{i1} \tag{9}$$

In the same manner as in the first "sub-triangle", eq. (9) after expanding, we obtain :

$$\begin{bmatrix} R^{i2}_{i1} & \lambda_2 t^{i2}_{i1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^{i2}_{j1} & \beta t^{i2}_{j1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^{j1}_{i1} & t^{j1}_{i1} \\ 0 & 1 \end{bmatrix} \tag{10}$$

λ_2 is the scale factor associated to the transformation T^{i2}_{i1} and β is the scale factor associated to the transformation T^{j1}_{i2}. The equation (9) is decoupled into rotation and translation terms and leads to :

$$R^{i2}_{i1} = R^{i2}_{j1} R^{j1}_{i1} \tag{11}$$

$$\lambda_2 t^{i2}_{i1} - \beta t^{i2}_{j1} = R^{i2}_{j1} t^{j1}_{i1} \tag{12}$$

In the "big-triangle" between the camera's positions C_{i0}, C_{i2}, and C_{j1}, Fig. 3, we can write the transformations as in the eq. (13). The transformation of the camera position C_{i2} to the camera coordinate system of C_{i0}, T^{i0}_{i2}, is equal to the transformation of the camera's position C_{j1} to the camera coordinate system of C_{i2}, T^{i0}_{j1}, transformed via the transformation of the camera's position of C_{i2} to the camera coordinate system of C_{j1}, T^{j1}_{i2}.

$$T^{i0}_{i2} = T^{i0}_{j1} T^{j1}_{i2} \tag{13}$$

In the same manner as in the "sub-triangles", eq. (13) after expanding, we obtain :

$$\begin{bmatrix} R^{i0}_{i2} & (\lambda_1 + \lambda_2) t^{i0}_{i2} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^{i0}_{j1} & \alpha t^{i0}_{j1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^{j1}_{i2} & \beta t^{j1}_{i2} \\ 0 & 1 \end{bmatrix} \tag{14}$$

After decoupling the rotation and translation terms, the translational part leads to the eq. (15).

$$\begin{bmatrix} t^{i0}_{i2} & t^{i0}_{i2} & -t^{i0}_{j1} & -R^{i0}_{j1} t^{j1}_{i2} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \alpha \\ \beta \end{bmatrix} = 0 \tag{15}$$

To summarize, the translation terms of the three "triangles" are expressed as :

$$\begin{cases} \lambda_1 t^{i0}_{i1} - \alpha t^{i0}_{j1} = R^{i0}_{j1} t^{j1}_{i1} \\ \lambda_2 t^{i2}_{i1} - \beta t^{i2}_{j1} = R^{i2}_{j1} t^{j1}_{i1} \\ \lambda_1 t^{i0}_{i2} + \lambda_2 t^{i0}_{i2} - \beta R^{i0}_{j1} t^{j1}_{i2} - \alpha t^{i0}_{j1} = 0 \end{cases} \tag{16}$$

To solve the above expression for the scale factors, the system of the previous equations is written as :

$$\begin{bmatrix} t^{i0}_{i1} & 0 & -t^{i0}_{j1} & 0 \\ 0 & t^{i2}_{i1} & 0 & -t^{i2}_{j1} \\ t^{i0}_{i2} & t^{i0}_{i2} & -t^{i0}_{j1} & -R^{i0}_{j1} t^{j1}_{i2} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} R^{i0}_{j1} t^{j1}_{i1} \\ R^{i2}_{j1} t^{j1}_{i1} \\ 0 \end{bmatrix} \tag{17}$$

The absolute scales factors can be derived by means of a linear least square (LS) model using the computed rotations and translations between the three frames and the known rotation and translation between the cameras (extrinsic calibration). The linear system can be written as this scheme:

$$A.X = B \tag{18}$$

Where X is the vector of the scale factors λ_1, λ_2, α and β.

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. The translation vectors of the same camera between three time steps C_{i0}, C_{i1}, and C_{i2} can be expressed in terms of unit vectors : t^{i0}_{i1}, t^{i1}_{i2}, and t^{i0}_{i2}. The following equation shows the relation between this unit translation vectors.

$$(\lambda_1 + \lambda_2) t^{i0}_{i2} = \lambda_1 t^{i0}_{i1} + \lambda_2 t^{i1}_{i2} \Leftrightarrow t^{i0}_{i2} = t^{i0}_{i1} = t^{i1}_{i2} \tag{19}$$

In the matrix A of the eq. (18), all the rotation matrices and the translation vectors are known parameters and the scale factors can be easily computed by solving the eq. (18).

After computing the relative poses and the absolute scale factors, the motion can be robustly estimated.

V. EXPERIMENTS

A. Simulation

The algorithm described in the previous sections is firstly applied to a simulation data. A 3D points cloud (randomly generated) of the scene is projected into two cameras (2D images) using the perspective camera model. For each camera, the poses are defined as a matrices in the world coordinate system. There are three pairs of such cameras resulting six different poses of ground truth. This pair of cameras is moving linearly between these positions and has a static transformation among each other to satisfy our assumptions. The intrinsic matrices of the cameras are defined and 3D points cloud of 100 points has been generated randomly. We obtain three frames by projecting 3D points into the camera coordinate systems : we obtain 100 2D points in each frame (without additive noise then with noise $\sigma = 0.2$). All points are in front of the cameras. Fig. 9 Shows the 3D modelisation of the scene.

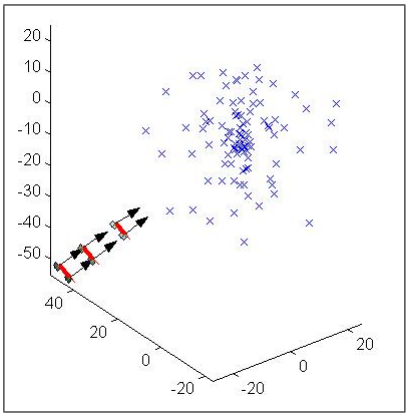

Fig. 5. simulated scene : 3D points cloud and two cameras in a linear segment motion.

From 2D-2D matching of the feature points, the euclidean transformations of the "triangles" are estimated. Using the extrinsic calibration pose between the cameras, the scale factors are estimated and compared to the ground truth value which are already calculated from the ground truth poses. The real scale factors are basically computed from the three generated poses (R and t of each camera in the world coordinate system) and the calibration poses between the cameras. In the simulation, we find a high motion estimation accuracy. The absolute scale results are presented in table I.

TABLE I

SIMULATION RESULTS : SCALE FACTORS COMPARISON FOR "TRIANGLE-BASED" METHOD (TM)AGAINST THE GROUND TRUTH (GT)

	λ_1	λ_2	α	β
GT	0.7291	1.4810	0.4379	2.3887
TM without noise	0.7291	1.4810	0.4379	2.3887
TM with noise ($\sigma = 0.2$)	0.8313	1.8344	0.3986	2.7781

In the case without noise, the scale factors calculated by our "triangle-based" method (TM) (λ_1, λ_2, α, and β) are equal to the scale factors of the ground truth (GT). In the case with additive noise, the results are very close. The inspection of the table reveals that the accuracy of our "triangle-based" method since the scales factors (GT and TM) is the same in the case without noise.

B. Real Data

In order to validate the "triangle-based" method, we verify the validity of our assumptions in the motion trajectory by comparing our results on a real world image sequence from KITTI dataset [13],[14] for two cameras against the GPS/INS ground truth. This dataset is captured from a VW station wagon for use in mobile robotics and autonomous driving research. The sensors used in our comparison are the gray scale stereo cameras and the GPS/IMU inertial navigation system. The recording platform use two PointGray Flea2 grayscale cameras, 1.4 Megapixels, and a OXTS RT3003 inertial and GPS RTK navigation [13],[14].

TABLE II

GROUND TRUTH (GT) AND "TRIANGLE-BASED" METHOD (TM) RESULTS

λ_1 TM	λ_1 GT	λ_2 TM	λ_2 GT
0.805 ± 0.533	0.706 ± 0.177	0.864 ± 0.509	0.705 ± 0.176

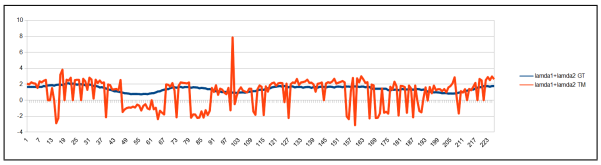

Fig. 6. $\lambda_1 + \lambda_2$ distribution for 223 "triangles" - the estimated scale factors are close to the scale factors of the ground truth. Some pikes are due to a low number of good matching (feature points)

The cameras are synchronized and we use only one image in every time step to apply our method. The relative poses and the absolute scale factors are computed for each set of 3 images. 2D-2D matching, Fig.7 , allows to estimate the essential matrix which is decomposed into rotation matrix and translation vector. Three calculated transformations and the extrinsic calibration are used to compute the scale factors as described in our method.

Results are shown in the table II and in the Fig. 6 which are uppermost mentioned. Fig. 6 shows the $\lambda_1 + \lambda_2$ distribution values for a sequence of 223 "triangles".

Fig. 8 plots the estimated and the real trajectories. The plot illustrates the trajectory of a curve motion. The trajectory shows the performance of our method in a curve motion and validate our assumptions about the linearity between consecutive frames of one camera. It is clearly shown that our method is very close to the real trajectory : The real trajectory represents the ground truth GPS positions and the estimated one represents the estimated positions by our "triangle-based" method.

A steady improvement of the estimated trajectory is observed in the Fig. 8. The trajectory is computed for a

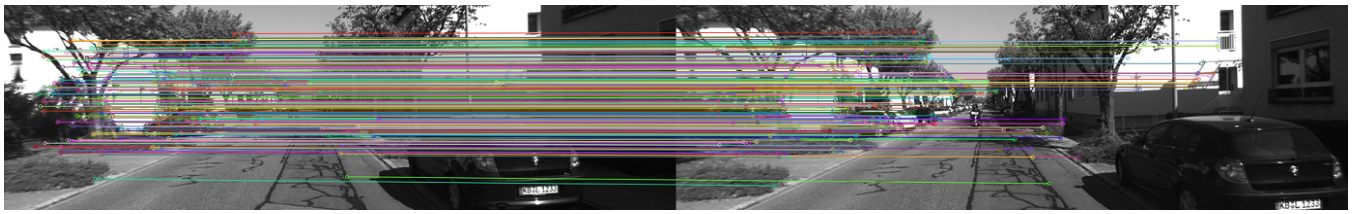

Fig. 7. Good matches using SURF detector and FREAK descriptor - Images from KITTI data base

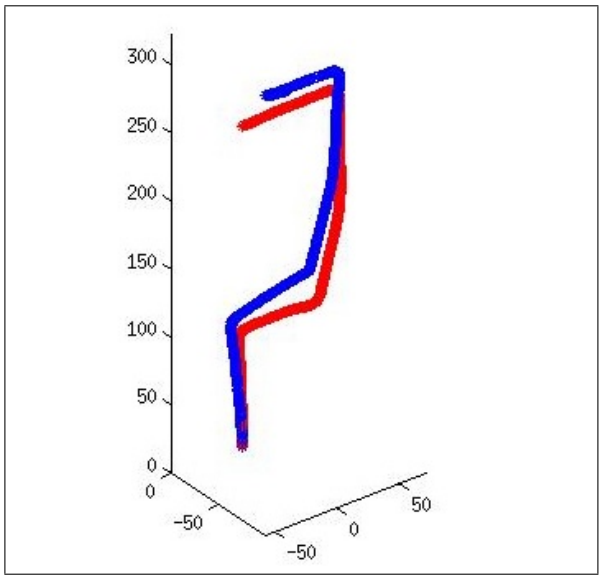

Fig. 8. Motion estimation : comparison between the "triangle-based" method estimation (red) and the GPS/INS ground truth poses (blue)- For this sequence of 549 frames, the estimated trajectory is very close to the real trajectory

sequence of 549 frames. The comparison among the trajectories shows some amounts of drift. We explain this difference by the quality of the feature matching distribution on each image. We found also two relevant errors in two situations. These errors are due to a low number of feature points. For instance, in some scenes there are very small clutters. As we calculate the current pose always in the system coordinate of the previous one, the errors are propagated and accumulated.

The estimated positions in a straight motion segment are very close to the ground truth positions. We consider these results to be encouraging for an unsynchronized multi camera setup. Also, we suppose that a some improvement could be obtained by applying a local bundle adjustment.

VI. CONCLUSIONS

In this paper, we have presented a novel motion estimation method which we called "triangle-based" method using an unsynchronized multi camera setup. The "triangle-based" method motion estimation assumes that the trajectory between two consecutive frames is approximated to be a linear segment (straight or smoothly curved). Our method requires an off line calibration knowledge and can be implemented on any mobile platform because of it flexibility and its real time implementation possibility. The unsynchronization will be useful by many applications such autonomous vehicle and many robotic tasks. The presented approach results improves high accuracy at the straight motion segment and still very close to the curve trajectory. Although we use a robust method, some errors can be spread due to the inaccuracy in the pose estimation part. These errors can be corrected by a local bundle adjustment.

ACKNOWLEDGMENT

This work is sponsored in part by DrAAcar project (N^{o} ANR-11-IS03-0003) and "Région Haute Normandie".

REFERENCES

[1] G. Lee, F. Fraundorfer, and M. Pollefeys. *Motion estimation for a self-driving car with a generalized camera.* p. 2746-2753, CVPR 2013.

[2] S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular slam based navigation for autonomous micro helicopters in gps denied environments. Journal of Field Robotics, Vol. 28 Issue 6, p. 854-874, 2011.

[3] Meilland, Maxime and Comport, Andrew Ian and Rives, Patrick. *Dense visual mapping of large scale environments for real-time localisation.* P. 4242-4248, IROS 2011.

[4] P. T. Furgale, P.l Newman, R. Triebel et al. *Toward Automated Driving in Cities using Close-to-Market Sensors, an overview of the V-Charge project.* p. 809-816. IV 2013.

[5] D. Nistr. *An efficient solution to the five-point relative pose problem.* In Pattern Analysis and Machine Intelligence, vol. 26, p. 756-777, 2004.

[6] H. Li and R. Hartley. *Five-point motion estimation made easy.* p. 630-633, ICPR 2006.

[7] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. *Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC.* p. 488-494, ICRA 2009.

[8] F. Fraundorfer, D. Scaramuzza, and M. Pollefeys. *A constricted bundle adjustment parameterization for relative scale estimation in visual odometry.* p. 1899-1904 ICRA, 2010.

[9] T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart. *Real-time 6d stereo visual odometry with non-overlapping fields of view.* p. 1529-1536, CVPR 2012.

[10] M. Svedman, L. Goncalves, N. Karlsson, M. E. Munich, and P. Pirjanian. *Structure from stereo vision using unsynchronized cameras for simultaneous localization and mapping.* p. 3069-3074. IROS, 2005.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. *Speeded-up robust features (surf).* In Computer Vision and Image Understanding, vol. 110, p. 346-359, 2008.

[12] A. Alahi, R. Ortiz, and P. Vandergheynst. *Freak : Fast retina keypoint.* p. 510-517. CVPR 2012.

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

[14] A. Geiger, P. Lenz, and R. Urtasun. *Are we ready for autonomous driving the kitti vision benchmark suite.* p. 3354-3361, CVPR 2012.

[15] C. Theobalt, H. Seidel, K. I. Kim, N. Hasler, C. Stoll, A. Elhayek. *Spatio-temporal motion tracking with unsynchronized cameras.* p. 1870-1877. CVPR, 2012.

[16] M. Bertozzi, L. Bombini, A. Broggi, et al. *The VisLab Intercontinental Autonomous Challenge: 13,000 km, 3 months, no driver.* Procs. 17th World Congress on ITS, 2010.

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