Asynchronous Structure from Motion at Scale

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

Noname manuscript No.
(will be inserted by the editor)

Asynchronous Structure from Motion at Scale

Rawia Mhiri · Safa Ouerghi · Rémi Boutteau · Pascal Vasseur · Stéphane Mousset · Abdelaziz Bensrhair

Received: date / Accepted: date

Abstract Vision systems that provide a 360-degree view are becoming increasingly common in today's vehicles. These systems are generally composed of several cameras pointing in different directions and rigidly connected to each other. The purpose of these systems is to provide driver assistance in the form of a display, for example by building a Bird's eye view around the vehicle for parking assistance. In this context, and for reasons of cost and ease of integration, such cameras are generally not synchronized. If non-synchronization is not a problem when it comes to display only, it poses significant issues for more complex computer vision applications (3D reconstruction, motion estimation, etc.). In this article, we propose to use a network of asynchronous cameras to estimate the motion of the vehicle and to find the 3D structure of the scene around it (for example for obstacle detection). Our method relies on the use of at least three images from two adjacent cameras. The poses of the cameras are independently estimated by conventional visual odometry algorithms. Then we show that it is possible to find the absolute scale factor by hypothesizing that the motion of the vehicle is smooth. The results are then refined through a local bundle adjustment on the scale factor and 3D points only. We evaluated our method under real conditions on the

R. Mhiri
Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, 76000 Rouen, France
E-mail: rawia.mhiri@insa-rouen.fr

S. Ouerghi
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: ouerghi@esigelec.fr

R. Boutteau
Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France
E-mail: boutteau@esigelec.fr

P. Vasseur
Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, 76000 Rouen, France
E-mail: pascal.vasseur@univ-rouen.fr

S. Mousset
Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, 76000 Rouen, France
E-mail: stephane.mousset@univ-rouen.fr

A. Bensrhair
Normandie Univ, UNIROUEN, UNIHAVRE, INSA Rouen, LITIS, 76000 Rouen, France
E-mail: abdelaziz.bensrhair@insa-rouen.fr

KITTI database, and we showed that our method can be generalized to a larger network of cameras thanks to a system developed in our lab.

Keywords Structure from Motion · Asynchronous cameras · Scale estimation

1 Introduction

In recent years, advanced driver assistance systems (ADAS) that were originally reserved for luxury vehicles have become available on models intended for the general public. These vehicles are now standard-equipped with many driver assistance features such as lane departure warning [24], blind spot monitoring [8], traffic sign recognition [39] [23], automatic parking or autonomous driving in traffic jams [22]. In this context of development of ADAS, the camera has become one of the most commonly used sensors because in addition to its low cost compared to other sensors such as lidars, the camera is a multi-function sensor that allows to develop new features on an already existing architecture. In the future, ADAS will offer more and more automated assistance and the development of fully autonomous vehicles will involve a complete and robust perception of the environment around the vehicle, while maintaining affordable costs for the automotive sector.

Monitoring the environment of a vehicle can be done with a very small number of cameras. An omnidirectional observation of the scene is indeed possible with the use of a catadioptric sensor, that is to say by combining a camera with a convex mirror [2]. Such systems have been widely studied in the robotic community [34] and many tests have also been made in the field of autonomous vehicles [36]. However, the use of these sensors to monitor the environment of a vehicle is not optimal since they must be placed on the roof of the vehicle to have a 360 degree view around it, and the roof can hide a part of the image corresponding to the environment close to the sensor. To mitigate this phenomenon, they must be sufficiently elevated, which makes their use in the automotive field not credible for both practical and aesthetic reasons.

Another possibility to obtain an omnidirectional vision is to use camera networks. There are several multi-camera systems in the literature and on the market, the best known being without undoubtedly the Ladybug®[1]. These systems, very compact, have several drawbacks. The first one is similar to catadioptric sensors, that is to say it must be placed on the roof and at a certain height to have an omnidirectional view. Secondly, their baseline is very weak and can be considered almost as a single view point, which does not allow to obtain 3D points by triangulation. Finally, these systems require a precise synchronization to obtain images of the scene simultaneously.

Automakers are integrating more and more cameras into their vehicles, especially to develop driver assistance systems (ADAS) such as the Nissan Around View® Monitor[2]. These systems are generally composed of four low-cost cameras that do not have a synchronization system. We are interested here in this type of configuration to propose a Structure from Motion (SfM) system using a network of non-synchronized cameras. The advantages of such a network are numerous. First of all, it is an inexpensive system that does not require additional wiring or synchronization hardware. It is also possible to associate cameras with different characteristics, and in this case the system no longer depends on the slowest camera. Unlike synchronized systems, non-synchronous networks are not susceptible to image loss that occurs quite frequently. In addition, the images being acquired continuously, the

[1] https://www.ptgrey.com/ladybug5-30-mp-usb-30-spherical-digital-video-camera-black

[2] http://www.nissan-global.com/EN/TECHNOLOGY/ MAGAZINE/around_view_monitor.html

bandwidth problems are reduced to a certain extent. This allows to have a localization at a higher frequency, that is to say whenever a new image is acquired. Finally, it is possible to add or remove one of the cameras in the system without affecting its proper functioning.

The originality and interest of our method is to study a non-synchronized multi-camera system to estimate the absolute scale factor for trajectory estimation or 3D reconstruction. This subject is not addressed in the literature as shown in the next section. Our main contribution is a new motion estimation algorithm, including the scale factor, that we have called triangle-based method. This method is based on the use of a triplet of images from two cameras with overlapping to estimate the relative poses between the images with conventional visual odometry algorithms. The absolute scale factors are then estimated by integrating a virtual pose of one of the cameras in the shape of the triangle formed by the image triplet. An optimization method was then proposed to improve the accuracy of the initial estimate. This optimization is based on a bundle adjustment (BA) on scale factors and 3D structure only applied on a sliding window.

2 Related work

3D reconstruction at scale from an embedded camera network requires the ability to establish epipolar geometry and can be performed according to several approaches based on prior knowledge about this system. If one considers the network to be calibrated, both intrinsically and extrinsically, as well as synchronized, with overlapping areas, then the classical techniques proposed in [16] can be applied to each set of images. It is also possible in this case to integrate consecutive images of the moving network in a multifocal tensor for a more accurate estimation [6].

For omnidirectional systems, [37] offers a SLAM (Simultaneous Localization and Mapping) based approach. In this work, the five cameras around the vehicle are synchronized and the interest points must be visible in the field of view of two cameras simultaneously in order to obtain the scale factor. In [1] and [43], a visualization system of the environment around a vehicle is also presented. These works concern more particularly the optimization at the hardware level with the use of SoC (System-on-Chip). The cameras in this system are synchronized by an external trigger, and the extrinsic calibration is rudimentary since the (hopefully miniature) vehicle is placed in the center of a pattern containing four squares of known positions and dimensions.

In the specific case of the vehicle, [32] presents a network of embedded cameras associated with a calibration method for a use dedicated to the risk analysis in an ADAS. Recently, a method dedicated to the Around View® Monitor system has also been proposed in [7] to perform the reconstruction of the environment. However, the synchronization of cameras, being of utmost importance to ensure the geometrical correctness, has been performed by hardware triggering.

If one or more of the three initial conditions (calibration, synchronization, overlapping) are released then dedicated methods are needed. Thus, in the calibrated and synchronized case but without overlapping between the cameras, it is possible to define a constraint on the rigid motion of the cameras to assemble the two separate views. An example of this type for a synchronous fisheye stereoscopic system is presented in [30]. To overcome the constraint of overlapping fields of view, a visual odometry algorithm is applied separately on each camera up to a scale factor. A linear solution is defined in order to merge the scale factors from the two estimates by imposing the known rigid transformation between the two sensors to finally find the absolute metric scale. More recently, [41] tackled the same problem by

proposing a procedure for initializing the scale with interesting results. All of these methods can more generally be related to the notion of generalized camera [29] [5] [18].

In the case where the extrinsic calibration is also missing, the methods generally propose to carry out both the reconstruction of the scene and the estimation of the calibration parameters. Thus, a vehicle-mounted multi-camera system was developed as part of the V-Charge project (Autonomous Valet Parking and Charging for e-Mobility) [11]. The extrinsic calibration is presented in [17] and [15]. The proposed approach is based on the use of vehicle odometry data in addition to visual data. A VO (Visual Odometry) algorithm followed by a bundle adjustment is used for each camera. The VO data thus obtained consequently have different scales for each camera. The interest points used for the VO are then triangulated using a first approximation of the camera-odometry transformation and the odometry data provided by the vehicle. A second bundle adjustment is then applied by fixing the camera poses and by optimizing the 3D points and the odometry-camera transformation. The accuracy is not yet sufficient since the reprojection errors remain high. To overcome this problem, the interest points are matched in the images of several cameras and loop closures are also performed when detected. A final bundle adjustment is then applied in order to optimize all the parameters. The disadvantage of this method is that it requires the use of additional data in addition to visual data.

In [27], the same problem is tackled by integrating temporal offsets due to the effect of rolling shutters. However, in these works, they use very restrictive assumptions. They assume identical cameras (operating at the same frequency), uniformly spatially distributed and placed close enough to approximate a single viewpoint. The approach is essentially based on a bundle adjustment optimizing all the parameters of the network and the 3D scene using the method initially proposed in [26] [19]. They seek to estimate the time shift only one time using the first images of a video sequence. This allows to synchronize cameras up to an image. A bundle adjustment is then performed since the images are this time considered as synchronized. To obtain the offset due to the rolling shutter, a second bundle adjustment is achieved by adding this additional parameter. Our method differs for several reasons: we do not have any assumption on the cameras, they can work with different frame-rates (and even not constant), moreover in this article they make a bundle adjustment a posteriori on the whole sequence, while we are doing an online bundle adjustment. Finally, the system studied in these works is considered to have a unique viewpoint, which means getting into the configuration of a monocular system.

In the case of unsynchronized cameras, the main focus has been on estimating the time offset between the different streams of video data [31] [42] [4]. Indeed, in a video stream with moving objects, objects move in the same way in all views and can be used to calculate the time offset. For an entire sequence of a dynamic scene observed by unsynchronized cameras, matching of primitives and synchronization can be post-processed and estimated. However, these approaches are not suitable in the case of the vehicle since the processing must be done online.

In [38], the approach is to simulate the missing image of an asynchronous system to compute the 3D structure using the odometry of the robot. The 2D points of the missing image are obtained by interpolation of the 2D points detected in the captured images. The approach is based on the linearity of the motion between two consecutive images of the same camera.

In all the works cited above, the absolute scale reconstruction implies necessarily either to add additional sensors, or to have synchronized cameras and points visible by at least two cameras.

3 Triangle-based method

3.1 Introduction

This section presents the method that we proposed to recover the motion of our asynchronous camera system. This method is based on three main hypotheses:

- Cameras must be mounted on a rigid system and have common fields of view. As the cameras are rigidly fixed, the transformation between each pair of camera can be known by extrinsic calibration. Common fields of view between two consecutive cameras allow to match interest points.
- Cameras must be calibrated off-line in order to know their intrinsic and extrinsic parameters. Extrinsic calibration will be used to find absolute scale factors and the intrinsic calibration will be used in the process of pose estimation and triangulation.
- The motion between two consecutive views is assumed to be linear and smooth, and is consequently approximated by a line segment. With this approximation we consider that translation vectors between three positions of the same camera are collinear. Indeed, since the acquisition frequencies of today's cameras are becoming higher and higher (20, 30, 40 frames per second and even more), the time interval between 3 images (i.e. positions) is not very large. For example, for a vehicle equipped with a 30 fps camera driven at 50 km/h (around 13m/s), the covered distance between 3 images is 1m30 (approximately one picture every 45 centimeters).

We separate the method into two parts: relative pose computation and absolute scale factor estimation. Relative poses of the cameras are estimated via SfM [28]. Then, the absolute scale factors are computed using the extrinsic calibration and the linearity assumption.

To simplify the explanation of our method and without loss of generality, we consider a system with only two cameras that acquire images at three different times. The first and second cameras, C_i and C_j, acquire images I_i and I_j respectively. The time notation is set as subscript attached to the camera's name. For example, the position of the camera C_i at time t_0 is designated by C_{i0}. The Euclidean transformation from the camera C_{i0} to the camera C_{j1} is given by $\mathbf{T}_{i0}^{j1}$. In the same way, the rotation matrix and the unit vector of the translation from the camera C_{i0} to the camera C_{j1} are denoted by $\mathbf{R}_{i0}^{j1}$ and $\mathbf{t}_{i0}^{j1}$.

3.2 Pose estimation

The Euclidean transformation between two relative poses of a camera can be described by a unitary translation vector $\mathbf{t}$, a rotation matrix $\mathbf{R}$ and a scale factor λ. The transformation between the two positions can be expressed as follows:

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

As our system is calibrated, we can estimate the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$ between two camera poses through the matching of interest points and by exploiting the epipolar geometry between the two views [16].

The triangle-based method relies on the estimation of three essential matrices between the images acquired at three different times: two images captured by the same camera (C_i for

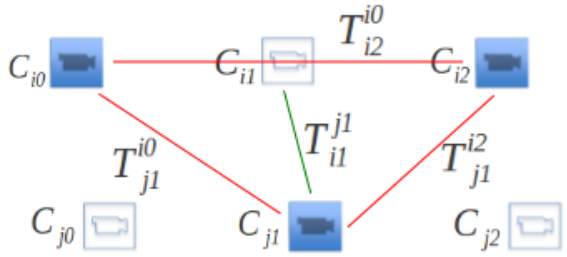


Fig. 1 The triangle-based method for two unsynchronized cameras: the red lines indicate the transformations obtained by SfM and the green lines indicate the rigid transformation resulting from the extrinsic calibration.

example) and one image by the other camera (C_j in this case). The triangle can be modeled as shown in Figure 1 which presents all the possible transformations between the three images forming the triangle. The transformations $\mathbf{T}^{i0}_{i2}$, $\mathbf{T}^{i0}_{j1}$ and $\mathbf{T}^{i2}_{j1}$ between the three images are determined by the 5-point algorithm [28]. The rigid transformation $\mathbf{T}^{j1}_{i1}$ is obtained from the offline extrinsic calibration process.

The first step of our algorithm is the extraction and the matching of interest points detected in the three images of the triangle. In our implementation, we use the FAST detector [33] and the BRIEF descriptor [3]. These steps are applied between the images I_{i0}, I_{i2} and I_{j1}.

The camera C_i passes through an intermediate position where it does not acquire an image due to the non-synchronization. This is the C_{i1} pose that can be estimated using the $\mathbf{T}^{j1}_{i1}$ transformation. To sum up, we use four transformations: three computed via SfM ($\mathbf{T}^{i0}_{i2}$, $\mathbf{T}^{i0}_{j1}$ and $\mathbf{T}^{i2}_{j1}$) and one rigid transformation ($\mathbf{T}^{j1}_{i1}$) resulting from the extrinsic calibration.

3.3 Scale factors estimation

Until then, the absolute scale factors are unknown between the poses of the cameras. With assumptions we made, it is possible to expressed the poses by four main transformations, as shown in the Figure 2. The three images (I_{i0}, I_{i2} and I_{j1}) form the so-called 'main triangle' between the poses of C_{i0}, C_{i2} and C_{j1}. The virtual pose of the camera C_{i1} can be considered as an intermediate position in this triangle. This position gives us two sub-triangles: the first one is formed by C_{i0}, C_{i1} and C_{j1} and the second is formed by C_{i1}, C_{i2} and C_{j1}.

As illustrated in Figure 2, in the first sub-triangle, the transformation $\mathbf{T}^{i0}_{i1}$ from the camera C_{i1} to the camera C_{i0} is equal to the transformation $\mathbf{T}^{i0}_{j1}$ from C_{j1} to C_{i0} multiplied by the transformation $\mathbf{T}^{j1}_{i1}$:

$$\mathbf{T}^{i0}_{i1} = \mathbf{T}^{i0}_{j1}\mathbf{T}^{j1}_{i1}. \tag{2}$$

Euclidean transformations are expressed in homogeneous coordinates, as shown in Equation (1). The rigid transformation $\mathbf{T}^{j1}_{i1}$ obtained by the extrinsic calibration is a scaled transformation, i.e. the scale factor of this transformation is known. Re-injecting Equation (1) in Equation (2), and introducing the two unknown scale factors λ_1 and α leads to:

$$\begin{bmatrix} \mathbf{R}^{i0}_{i1} & \lambda_1 \mathbf{t}^{i0}_{i1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{i0}_{j1} & \alpha \mathbf{t}^{i0}_{j1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^{j1}_{i1} & \mathbf{t}^{j1}_{i1} \\ 0 & 1 \end{bmatrix}, \tag{3}$$

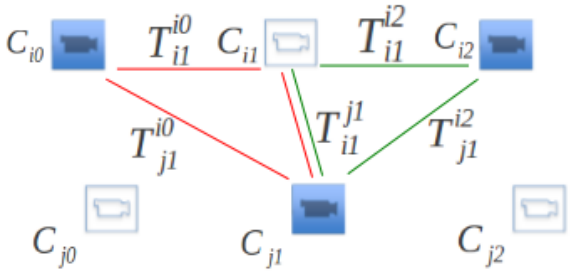


Fig. 2 The first sub-triangle between the cameras (C_{i0}, C_{i1} and C_{j1}) and the second sub-triangle between the cameras (C_{i1}, C_{i2} and C_{j1}).

where λ_1 is the scale factor associated to the $\mathbf{T}^{i0}_{i1}$ transform and α is the scale factor associated to the $\mathbf{T}^{i0}_{j1}$ transformation.

From (3), we can obtain the two following equations:

$$\mathbf{R}^{i0}_{i1} = \mathbf{R}^{i0}_{j1}\mathbf{R}^{j1}_{i1}, \tag{4}$$

and

$$\lambda_1 \mathbf{t}^{i0}_{i1} - \alpha \mathbf{t}^{i0}_{j1} = \mathbf{R}^{i0}_{j1}\mathbf{t}^{j1}_{i1}. \tag{5}$$

Moreover, Equation (5) can be written:

$$\begin{bmatrix} \mathbf{t}^{i0}_{i1} & -\mathbf{t}^{i0}_{j1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \alpha \end{bmatrix} = \mathbf{R}^{i0}_{j1}\mathbf{t}^{j1}_{i1}. \tag{6}$$

As shown in Figure 2, in the second sub-triangle, the transformation $\mathbf{T}^{i2}_{i1}$ from C_{i1} to C_{i2}, is equal to the transformation $\mathbf{T}^{i2}_{j1}$ from camera C_{j1} to camera C_{i2} multiplied by the rigid transform $\mathbf{T}^{j1}_{i1}$ from camera C_{i1} to camera C_{j1}:

$$\mathbf{T}^{i2}_{i1} = \mathbf{T}^{i2}_{j1}\mathbf{T}^{j1}_{i1}. \tag{7}$$

In the same way as in the first sub-triangle, Equation (7) is developed to obtain:

$$\begin{bmatrix} \mathbf{R}^{i2}_{i1} & \lambda_2 \mathbf{t}^{i2}_{i1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}^{i2}_{j1} & \beta \mathbf{t}^{i2}_{j1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}^{j1}_{i1} & \mathbf{t}^{j1}_{i1} \\ 0 & 1 \end{bmatrix}. \tag{8}$$

λ_2 represents the scale factor associated with the transformation $\mathbf{T}^{i2}_{i1}$. β represents the scale factor associated with the transformation $\mathbf{T}^{j1}_{i2}$.

Let us separate in Equation (8) the rotation and translation terms, we then obtain:

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

and

$$\lambda_2 \mathbf{t}^{i2}_{i1} - \beta t^{i2}_{j1} = \mathbf{R}^{i2}_{j1}\mathbf{t}^{j1}_{i1}. \tag{10}$$

In the main triangle formed by the triplet C_{i0}, C_{i2} and C_{j1}, the Euclidean transformations between the poses can be expressed as in Equation (11). The transformation $\mathbf{T}^{i0}_{i2}$ from C_{i2} to C_{i0} is equal to the transformation $\mathbf{T}^{i0}_{j1}$ from C_{j1} to C_{i0} multiplied by the transformation $\mathbf{T}^{j1}_{i2}$ from C_{i2} to C_{j1}:

$$\mathbf{T}_{i2}^{i0} = \mathbf{T}_{j1}^{i0}\mathbf{T}_{i2}^{j1}. \tag{11}$$

As for sub-triangles, and using the linearity assumption, Equation (11) becomes:

$$\begin{bmatrix} \mathbf{R}_{i2}^{i^0} & (\lambda_1+\lambda_2)\mathbf{t}_{i2}^{i^0} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{j1}^{i^0} & \alpha\mathbf{t}_{j1}^{i^0} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{i2}^{j1} & \beta\mathbf{t}_{i2}^{j1} \\ 0 & 1 \end{bmatrix}. \tag{12}$$

After the development of Equation (12), we separate rotation and translation terms. The translation leads to Equation (13):

$$\begin{bmatrix} \mathbf{t}_{i2}^{i0} & \mathbf{t}_{i2}^{i0} & -\mathbf{t}_{j1}^{i0} & -\mathbf{R}_{j1}^{i0}\mathbf{t}_{i2}^{j1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \alpha \\ \beta \end{bmatrix} = \mathbf{0} \tag{13}$$

Consequently, translation-based equations for the three triangles can be expressed by the following system:

$$\begin{cases} \lambda_1\mathbf{t}_{i_1}^{i^0} - \alpha\mathbf{t}_{j_1}^{i^0} = \mathbf{R}_{j_1}^{i^0}\mathbf{t}_{i_1}^{j^1} \\ \lambda_2\mathbf{t}_{i_1}^{i^2} - \beta\mathbf{t}_{j_1}^{i^2} = \mathbf{R}_{j_1}^{i^2}\mathbf{t}_{i_1}^{j^1} \\ \lambda_1\mathbf{t}_{i_2}^{i^0} + \lambda_2\mathbf{t}_{i_2}^{i^0} - \beta\mathbf{R}_{j_1}^{i^0}\mathbf{t}_{i_2}^{j^1} - \alpha\mathbf{t}_{j_1}^{i^0} = \mathbf{0}. \end{cases} \tag{14}$$

To solve these equations and obtain scale factors, we can write this system as follows:

$$\begin{bmatrix} \mathbf{t}_{i1}^{i0} & 0 & -\mathbf{t}_{j1}^{i0} & 0 \\ 0 & \mathbf{t}_{i1}^{i2} & 0 & -\mathbf{t}_{j1}^{i2} \\ \mathbf{t}_{i2}^{i0} & \mathbf{t}_{i2}^{i0} & -\mathbf{t}_{j1}^{i0} & -\mathbf{R}_{j1}^{i0}\mathbf{t}_{i2}^{j1} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{j1}^{i0}\mathbf{t}_{i1}^{j1} \\ \mathbf{R}_{j1}^{i2}\mathbf{t}_{i1}^{j1} \\ \mathbf{0} \end{bmatrix} \tag{15}$$

Absolute scale factors can be estimated by solving Equation (15) in the least squares sense since this equation can be written as a linear system of the form:

$$\mathbf{AX} = \mathbf{B} \tag{16}$$

where $\mathbf{X}$ is the vector composed of scale factors λ_1, λ_2, α and β. It is consequently possible to estimate all the relative poses, including the scale factors with our method.

4 Bundle adjustment optimization

The initial estimate of scale factors is not sufficiently accurate because of the assumptions underlying the triangle-based method, in particular for trajectories with high curvatures and/or high velocities. For this reason, the initial estimate can not be used directly in navigation applications, obstacle detection, etc.

In this section, we propose to perform a local BA, i.e. a BA applied to a limited number of views. In our case, the cameras are supposed to be calibrated, so we consider that intrinsic parameters do not have to be refined. Parameters to be optimized by our algorithm are thus poses of the system and 3D points coordinates.

Bundle adjustment consists in simultaneously refining 3D coordinates describing the scene geometry and camera parameters (poses, eventually intrinsic parameters). The criterion usually minimized in a BA algorithm is the reprojection error, i.e. the error measured between the points observed in the image and the estimated projection of their corresponding 3D points (which depends on parameters to be estimated). In the case of a standard pinhole camera, the projection $\mathbf{x}$ of a 3D point $\mathbf{X}$ whose coordinates are expressed in homogeneous coordinates in a world frame is given by:

$$\mathbf{x} \sim [\mathbf{K}\,\mathbf{0}] \underbrace{\begin{bmatrix} \mathbf{R} & s\,\mathbf{t} \\ \mathbf{0}^\intercal & 1 \end{bmatrix}}_{\mathbf{T}_{world}^{cam}} \mathbf{X} = \mathbf{P}\mathbf{X}, \tag{17}$$

where $\mathbf{K}$ is the matrix of intrinsic parameters and $\mathbf{T}_{world}^{cam}$ is the transformation matrix from the world frame to the camera frame. The resulting matrix $\mathbf{P}$ is the projection matrix of the camera.

In the general case, the BA can optimize intrinsic parameters (the calibration matrix K), extrinsic parameters (the rotation and the translation of the camera with respect to the world frame) and 3D points. The problem must therefore be solved by minimizing the reprojection error as a nonlinear least squares optimization problem, for example using the Levenberg-Marquardt algorithm [21] (see Algorithm 1).

4.1 Problem formulation

Let $\widehat{\mathbf{X}}_i$ be the estimated coordinates of the i$^{\text{th}}$ point and n the number of points. Let $\widehat{\mathbf{P}}^j$ be the estimated projection matrix of the j$^{\text{th}}$ camera and m the number of cameras. $\mathbf{x}_i^j$ is the extracted image point corresponding to the i$^{\text{th}}$ point in the image of the j$^{\text{th}}$ camera. The cost function to be minimized can thus be written:

$$\min_{\widehat{\mathbf{X}}_i, \widehat{\mathbf{P}}^j} \sum_{j=1}^{m} \sum_{i=1}^{n} d(\widehat{\mathbf{P}}^j \widehat{\mathbf{X}}_i,\, \mathbf{x}_i^j)^2 \tag{18}$$

where $d()$ is the Euclidean distance. The minimization of the cost function is performed by the Levenberg-Marquardt algorithm. This algorithm consists essentially in calculating the Jacobian matrix and optimizing the system of equations iteratively. The key step of this algorithm is the resolution of the augmented normal equation:

$$(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\Delta = -\mathbf{J}^T\mathbf{e}, \tag{19}$$

where $\mathbf{J}$ is the jacobian matrix of the projection function, λ is a scalar varying from iteration to iteration, Δ is the increment vector of the estimates, $\mathbf{e}$ is the vector of reprojection errors, and $\mathbf{I}$ is the identity matrix.

4.2 Bundle Adjustment related work

BA has reached a certain maturation in the litterature [9] [40]. BA has been widely studied for visual odometry and SfM applications, but there is, to the best of our knowledge, no work that has been done on asynchronous systems.

Algorithm 1 Pseudo-code of the Levenberg-Marquardt algorithm

$i \leftarrow 0$
$\lambda \leftarrow 0.001$
Evaluation of $\|e(\mathbf{P_0})\|$
while $i <$ MAX_ITERATIONS and $\|e(P_i)\| >$ threshold **do**
  Resolution of the augmented normal equation:
  $(\mathbf{J}^T\mathbf{J} + \lambda\mathbf{I})\Delta = -\mathbf{J}^T\mathbf{e}$
  Evaluation of the new parameters vector $\mathbf{P_{i+1}} = \mathbf{P_i} + \Delta$:
  if $\|e(\mathbf{P_{i+1}})\| \geq \|e(\mathbf{P_i})\|$ **then**
    $\lambda \leftarrow 10\lambda$
  else
    $\lambda \leftarrow \lambda/10$, $P_{i+1} \leftarrow P_i + \Delta$
  end if
  $i \leftarrow i + 1$
end while

In the method presented by Engels et al. [9], a windowed bundle adjustment has been introduced to locally optimize camera poses. This method was presented for a sequence of an object on a turntable and consists of optimizing all parameters. On a larger scale, Mouragnon et al. [25] have also presented a local BA adapted for real-time applications and for long sequences obtained from calibrated cameras.

In the case of calibrated cameras, intrinsic parameters are known and are not to be optimized. The difference between our method and classical BA algorithms applied to calibrated cameras lies in the number of parameters to be optimized. On the one hand, a classical BA algorithm aims at optimizing 6 parameters per camera (3 parameters for the rotation using the Rodrigues parameterization and 3 parameters for the translation), and 3 parameters per 3D point. On the other hand, our algorithm aims at optimizing only scale factors and 3D points and thus 1 parameter per camera and 3 parameters per 3D point.

In the method presented by Fraundorfer et al. [10], a constrained BA was presented for a visual odometry problem from a single camera mounted on a vehicle. The main difference between this method and conventional BA methods is the separation of the relative motion estimation, presented in [35], and the scale estimation. Indeed, authors emphasize the coherent estimation of the scale by optimizing only distances between the neighboring cameras (the relative scale factors). Rotations and directions of translations initially estimated by the 1-point RANSAC (Random Sample Consensus) method are considered set and 3D points are calculated at each iteration of the optimization process. This is a global BA that optimizes all scales of a trajectory. Differences between this method and our method are:

- the camera configuration: the method of Fraundorfer et al. [10] is applied to a monocular system and our method is applied to an asynchronous camera network,
- authors propose an algorithm based on the hypothesis of circular motion whereas ours is based on a hypothesis of rectilinear motion,
- we optimize scale factors and 3D points simultaneously whereas the method proposed by Fraundorfer et al. [10] only optimizes scale factors. They compute 3D points for every iteration with the new scale factors.

4.3 Optimization of scale factors and 3D structure with our local Bundle Adjustment

As we suppose $\mathbf{R}$ and $\mathbf{t}$ being known, and we are only optimizing the scale factor s and the 3D coordinates of a point X, the projection described in Equation (17) can be written as a function F depending on s and $\mathbf{X}$:

$$F(\mathbf{X}, s) = \mathbf{PX}. \tag{20}$$

In our method, the Jacobian matrix $\mathbf{J}$ is computed by differentiating the projection function F with respect to the scale factor s and the 3D point $\mathbf{X}$ only. Let $\mathbf{J_X}$ be the Jacobian matrix (2×3 matrix) of F with respect to the point $\mathbf{X}$, and $\mathbf{J_S}$ be the Jacobian matrix (2×1 matrix) of F with respect to the scale factor s, then:

$$\mathbf{J_X} = \left[\frac{\partial F}{\partial \mathbf{X}}\right]_{2\times 3} \text{ and } \mathbf{J_S} = \left[\frac{\partial F}{\partial s}\right]_{2\times 1}. \tag{21}$$

For each 3D point and camera pose, the $\mathbf{J_X}$ and $\mathbf{J_S}$ matrices are computed for the considered sliding window. The resulting Jacobian $\mathbf{J}$ has a sparse structure as shown in Figure 3. If we consider m cameras and n 3D points, the Jacobian is a $(2 \cdot n \cdot m) \times (m + 3 \cdot n)$ matrix.

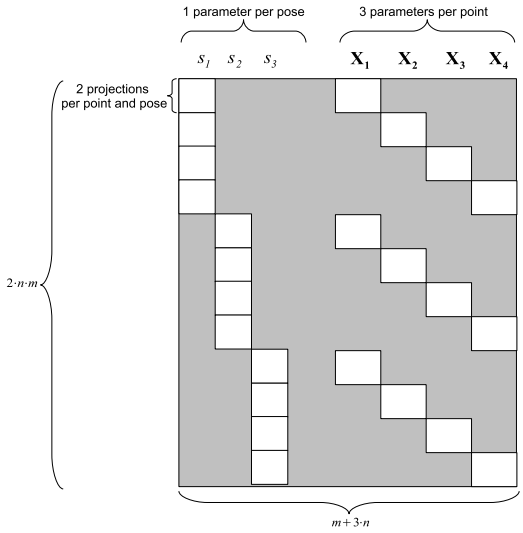


Fig. 3 Structure of the Jacobian matrix for 3 poses and 4 points. Zero entries in the matrix are shown in gray.

Once the Jacobian matrix has been calculated, the Leven-berg-Marquardt algorithm is implemented as described in Algorithm 1.

5 Results and Discussion

5.1 Evaluation of the triangle-based method

In order to validate the triangles method, we check the validity of our hypotheses for a real world sequence on the KITTI database [12] [13]. We compare the results of our method for

a sequence from two cameras perfectly synchronized with the ground truth obtained by the measurements of a GPS/INS. The KITTI dataset is a public database that has been acquired from an instrumented vehicle. We use stereo sequences by taking a single image at each time step to simulate the desynchronization. In other words, we use the even images for the left camera, and the odd images for the right camera, which allows us to have non-synchronized images. Relative poses and scale factors are computed for each set of three images forming a main triangle.

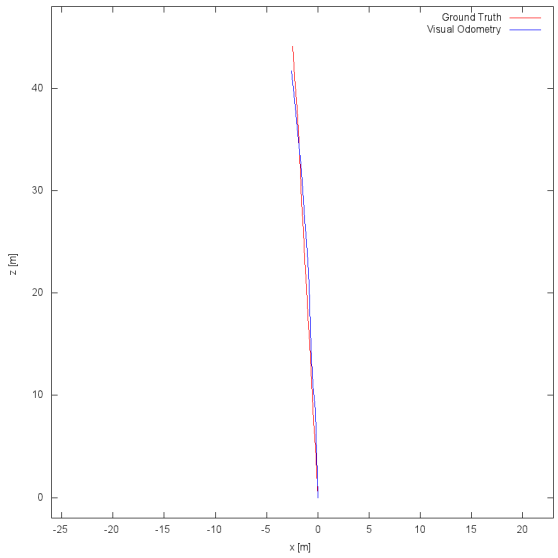
For each serie of three images (three moments), interest points are extracted using the FAST detector and described with the BRIEF descriptor. Then, the robust 5-point algorithm allows to simultaneously estimate the essential matrix as well the inlier pair of points. Rotations and relative translations are obtained from the decomposition of estimated essential matrices. Interest points considered as inliers are then triangulated in order to estimate 3D points coordinates. The system of equations (15) is solved to compute the absolute scale factors λ_1, λ_2, α and β.

To evaluate our method, we compare the results of our estimation with the ground truth obtained by the GPS/INS system. In this article, we do not compare our results with other methods of the state of the art since the global conditions are drastically different and more challenging in our case. Indeed, our approach is based on unsynchronized images. We therefore have less information than the synchronous stereoscopic case. We also do not compare our method to monocular methods because we are not in the same configuration. In the monocular case, the motion of a camera between two consecutive time steps generally presents small changes of field of view even at high speed. However, the images from several unsynchronized cameras undergo larger field of view changes.

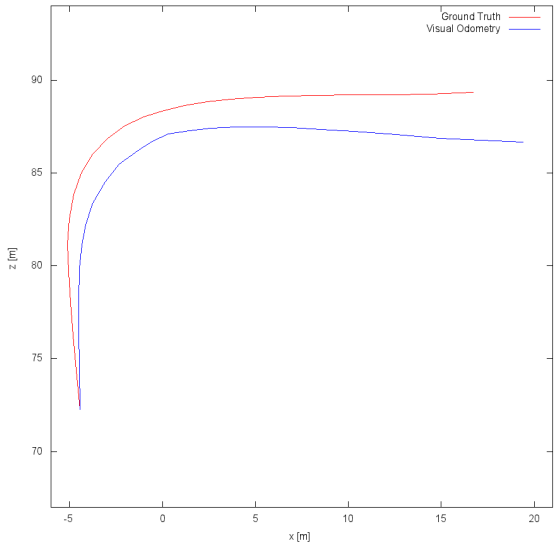
For linear trajectories, the results are very close to the ground truth as shown in Figure 4(a). Figure 4(b) illustrates the estimated trajectory in a turn. Even in this configuration (a rotation of about 90 degrees), the triangle-based method gives satisfactory results. Indeed, a vehicle usually does not move fast in turns. In addition, recent cameras have a sufficiently high frame rate. Taking into account these two points, the approximation of the motion between two images of the same camera as a line segment remains valid. The trajectory in a turn can therefore be decomposed as a piecewise linear trajectory between two images of a camera. The worst case would be a pure rotation, which can be ruled out since the car is a nonholonomic vehicle that can not perform this kind of movement. Moreover, even in case of pure rotation around one of the cameras, the other cameras of the system would undergo a displacement because of the lever arms between the different cameras. For all these reasons, the assumption of linearity remains valid, which is demonstrated by all the experiments that we have conducted.

Figure 7 shows the mean results on the eleven sequences of the KITTI dataset in terms of translation and rotation errors. We obtained rotational errors between 0.041 and 0.015 degrees per meter for sub-sequences between 100 and 800 meters (see Figure 7(a)) and between 0.14 and 0.02 degrees per meter for speeds up to 90 kilometers per hour. For translation, the average errors are between 7 and 9% for sub-sequences from 100 to 800 meters and between 5 and 22% for speeds upto 90 km/h.

We notice that rotational errors are higher for short sub-sequences and for slow speeds. Translation errors grow by about 2% for the longest sub-sequences. However, the other methods that perform optimization, such as bundle adjustment, get decreasing translation errors as a function of the lengths of the sub-sequences. The two extra percent is certainly due to the accumulation of errors. We explain these errors by the computation of scale fac-



(a) Trajectory in a straight line.

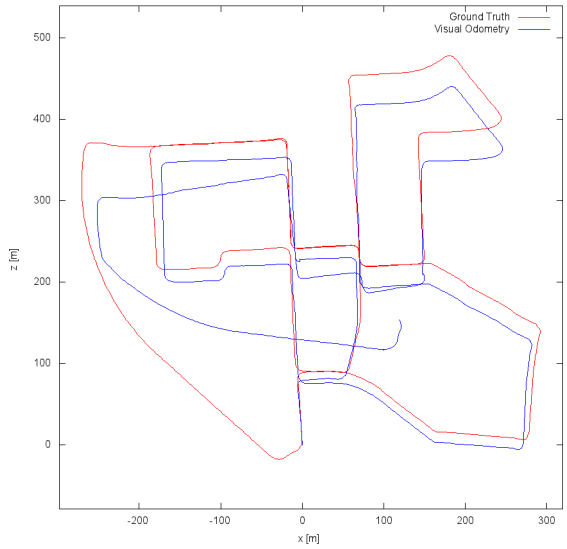


(b) Trajectory in a turn.

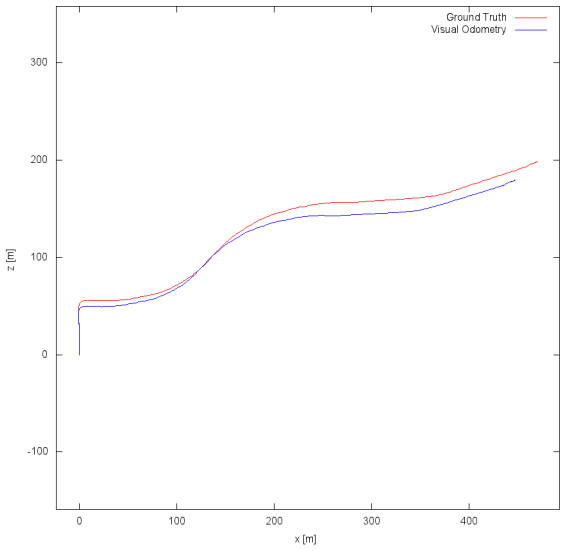
Fig. 4 Triangle-based trajectory estimation (in blue) and ground truth trajectory (in red) in two specific cases: (a) straight line, (b) turn.

tors. Small errors due to the approximation of a linear trajectory accumulate for the longest sub-sequences.

Sequence 0 is a sequence of 4540 images in urban scenes. The scenes are mainly composed of buildings, trees and cars. The distance traveled in this sequence is 2.232 kilometers, the mean distance between two images is 0.49 meters and the frame rate is 10 frames per



Sequence 0.

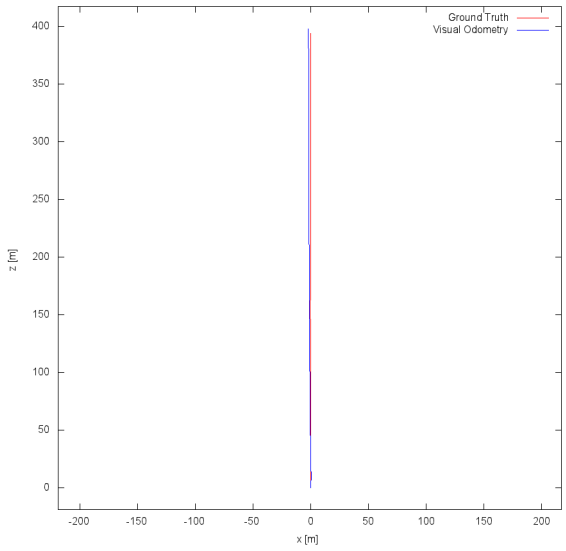


Sequence 3.

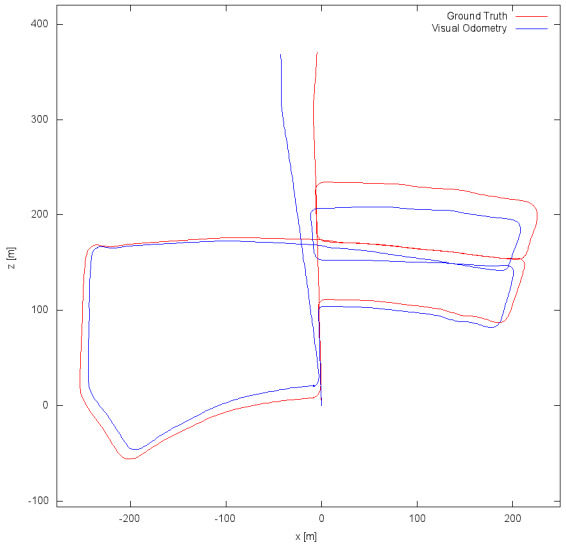
Fig. 5 The trajectories of sequences 0 and 3 of the KITTI dataset. The estimated trajectory is plotted in blue, the ground truth trajectory in red.

second. The mean translation error is 5.1% with the triangle-based method. As shown in Figure 5(a), despite the length of the sequence and the fact that it contains several turns, our method gives good results. The trajectory shows nevertheless a drift towards the end.

Sequence 3 (see Figure 5(b)) is a sequence of 800 images acquired over a distance of approximately 200 meters. This sequence presents scenes mainly composed of vegetation



(a) Sequence 4.

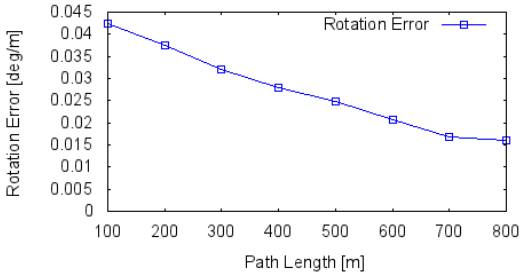


(b) Sequence 5.

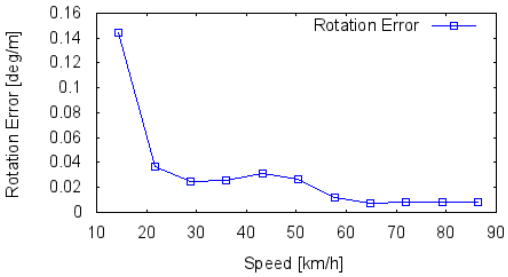
Fig. 6 The trajectories of sequences 4 and 5 of the KITTI dataset. The estimated trajectory is plotted in blue, the ground truth trajectory in red.

and few buildings. The resulting trajectory is close to the ground truth with average errors around 5% for translation and 0.006 degrees per meter for rotation.

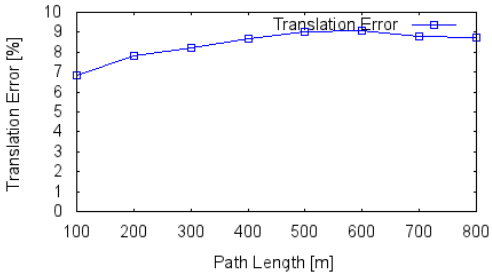
Our method provides the best results on Sequence 4 as shown in Figure 6(a). This sequence is composed of 270 images acquired over a distance of approximately 400 meters. This sequence represents a 4 lanes road that contains ground markings as well as traffic lights and intersections. Despite the abundance of vegetation, buildings, and cars moving



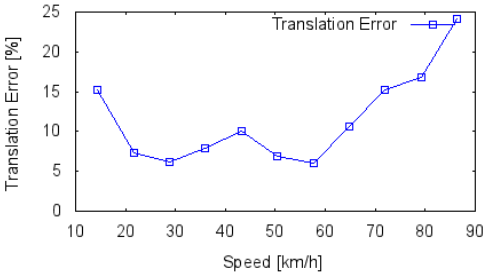
(a)



(b)



(c)



(d)

Fig. 7 Evaluation on the eleven sequences of the KITTI dataset provided with ground truth.

in the opposite direction, this scene is strongly structured due to markings. The trajectory obtained is very close to the ground truth with a mean error around 1.2% for translation and 0.006 degrees per meter for rotation.

Sequence 5 contains scenes very similar to Sequence 0: buildings, cars, etc. The scenes of this trajectory are very diversified. This is a sequence of 2760 images acquired over a distance of approximately 1050 meters. The estimated trajectory is close to the ground truth with a mean error around 4% for translation and 0.01 degrees per meter for rotation.

5.2 Evaluation of the BA optimization

5.2.1 Interest and evaluation of an optimal BA

In this section, we present the results of the scale factor optimization method described in the section 4. The algorithm is also applied on a sequence of real images of the KITTI database. We simulate the asynchronous aspect in the same way as for the experiments described previously. The sliding window BA is applied on two consecutive triangles, i.e. five images.

The 3D points and the absolute camera poses are initially estimated by the triangles method. To apply the BA, we express poses and 3D points in the first camera frame of the sliding window. Then, we apply the Levenberg-Marquardt algorithm.

The results obtained are presented for a part of the sequence 0 of the KITTI dataset, where there is both a rectilinear and a turning trajectory. First, we validate the method on a perfect estimate: 3D points are triangulated using the ground truth from the GPS/INS system which is synchronized with the images. Then, we apply the BA algorithm on this data. We name this test "Optimal Bundle Adjustment" since it will serve as a reference to which we will then compare the BA applied on the triangle-based method.

For the evaluation of the optimal BA, we calculate the ratio of the scale factors before and after applying the BA to the ground truth data as described in the equations (22) and (23). Scale factors are the norms of translation vectors. The obtained ratios are very close to 1, which means that the whole process and data are very close to reality. The low errors are probably due to low inaccuracy in the calibration and/or the synchronization of the sensors, the extraction of interest points, as well as the triangulation process.

$$\text{Ratio before BA} = \frac{\text{Scale factor estimated before BA}}{\text{Scale factor from the ground truth}} \tag{22}$$

$$\text{Ratio after BA} = \frac{\text{Scale factor estimated after BA}}{\text{Scale factor after optimal BA}} \tag{23}$$

To evaluate the performance of the approach, we add a Gaussian noise ($\sigma = 0.01$) to the ground truth (before BA). Noise is directly added to the scale factor values (before BA) because we are trying to optimize these parameters. 3D points are then calculated from the noisy poses before the optimization of all parameters. The scale factors evaluated before BA are therefore different from the scale factors of the ground truth.

The obtained results are satisfactory since the trajectory obtained after BA is almost the same than the one obtained by GPS. The scale factors ratios before and after BA are presented in Table 1. Scale 1 is the scale factor of the second camera pose in the sliding window in the frame of the first camera, Scale 2 is the scale factor of the third camera pose in the sliding window. It is the same for scales 3 and 4. The calculated ratios are close to 1, which means that our algorithm gives very accurate results.

Table 1 Ratios before and after BA applied to the ground truth with a gaussian 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

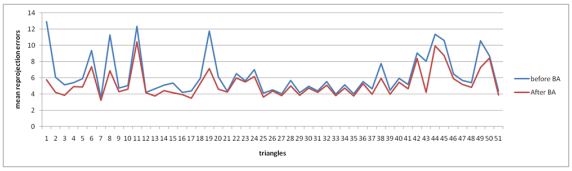


Fig. 8 Mean reprojection errors for 52 triangles before and after BA applied to the data estimated by the triangles method.

In summary, the errors obtained are very small and due to several reasons: the inaccuracy of the interest points detectors, the matching errors, the triangulation process, and the reprojection in images. By introducing noise in the scale factors, BA minimizes reprojection errors to improve the estimated scale factors and 3D points. When the poses are perfect (i.e. from the ground truth), some errors still remain. These errors are probably due to low inaccuracy in the calibration and/or the synchronization of the sensors, or even to inaccuracy in the GPS/INS system. That is why we compare the results obtained by the triangle-based method to an optimal BA.

5.2.2 Quantitative evaluation of the BA applied to the triangle-based method

In this section, we present the results of the BA applied to the triangle-based method data.

Figure 8 shows the distribution of mean reprojection errors before and after BA for a sequence of 200 images. Reprojection errors dropped significantly after BA. Figure 9 illustrates the reprojection errors of the 3D points in five cameras of a sliding window. Reprojection errors are represented in different colors for each camera.

We calculate the ratios in the same way as in equations (22) and (23) with the scale factors estimated by the triangle method before and after BA. Our results are summarized in Table 2. The trajectories obtained are presented in Figure (10). This figure shows that the trajectory optimized by the proposed method is closer to the ground truth than the trajectory initially estimated by the triangles method (before the BA).

Table 2 Ratios before and after BA applied to the 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) Reprojection errors before BA.

(b) Reprojection errors after BA.

Fig. 9 Example of reprojection errors of the 3D points before and after BA, each color refers to the reprojection errors in a camera of the sliding window (5 images).

5.3 Qualitative evaluation of our method on a more complex system

To show that our method can be generalized to a system with more cameras, we have developed a system consisting of 5 cameras rigidly mounted on the roof of a vehicle.

The cameras we use to carry out our experiments are Basler Ace 1600 (GigE, CCD 1/1.8", 1624x1234) cameras. These cameras are equipped with 6mm lenses, which leads to a 60° field of view. The system covers a field of view of 180° in front of the vehicle since each camera pair has a half-frame overlapping area (see Figure 11). This experiment also allowed us to test our algorithms with a narrow overlapping area between two images (50%) compared to the KITTI dataset where it is 100% since the images are rectified and resized. We think that 50% is a good value that we should not go below as, if so, the number of matched points may be not enough, especially during fast motions (fast rotations for example). The experimental platform was mounted on a vehicle and the dataset was collected during the driving of the car over a distance of about 350 meters around our campus. In this experiment, cameras have no synchronization system (e.g. hardware trigger) and the images are grabbed on the fly, leading to random times between the images of different cameras.

Figure 12 shows some images captured during our experiments. For each new image, we start by checking for possible transformations. The primitives were extracted by FAST and described by BRIEF and the relative transformations were estimated using the 5-point algorithm. A relative transformation is valid when the number of matches obtained is greater than a threshold of 50 points. Then, for each pose, the possible triangles are checked.

The triangle-based method followed by our BA is applied in each triangle to calculate the absolute scale factors λ_1, λ_2, α and β and the 3D structure of the scene. The estimated trajectory plotted in Figure 13 shows qualitatively interesting results since it matches the road, as well in the straight lines as in the turns, and with the correct scale.

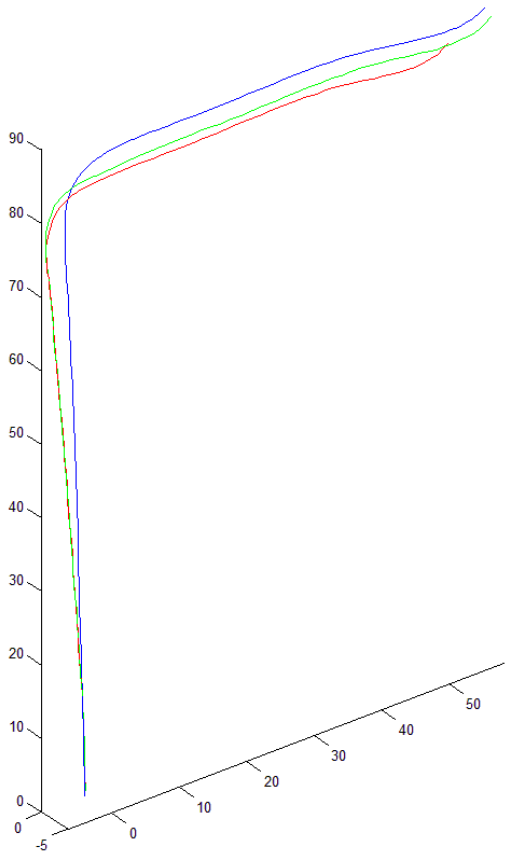


Fig. 10 Trajectories obtained from 200 images: triangle-based method in red, ground truth in blue, BA in green.

Fig. 11 Overlapping area between two images of our system. Each camera has a half-frame overlapping area.

Fig. 12 Samples of images from the non-synchronized camera network.

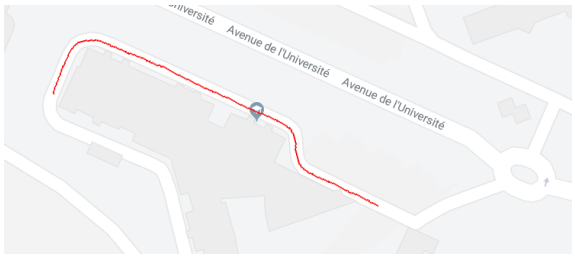


Fig. 13 The estimated trajectory in the real experiment.

5.4 Comparison with a stereoscopic system

As we will discuss in the next section, it is not fair to compare our system to a conventional stereoscopic system because we have an additional difficulty introduced by non-synchronization. We still did this comparison to show the importance of our system.

For the first experiment, we have compared the results of our method with the results of a stereovision-based scaling method in both the synchronized and non-synchronized cases. The results are shown in Figure 14. The green curve is the ground truth trajectory. The red curve is the result of our method. The black curve is the result using a synchronized stereoscopic system. To obtain these results, we estimate the motion of the left camera using the 5-point algorithm and we introduce the scale factor obtained by stereovision. To do so, points are triangulated at times t and t+1 to obtain several 3D points. The EPnP algorithm [20] is then used to recover the full motion (including the scale factor) and this scale factor is reinjected in the result of the 5-pt algorithm. As we can see in this figure, the stereoscopic case gives better results than ours. This is normal since in our case, we deal with an additional difficulty as we process non-synchronized images and with a smaller number of images (we only use one image out of two). We have also evaluated what would happen if we use stereovision algorithms on images that are not synchronized. For that, we have estimated the trajectory between the even images of the left camera with the 5-point algorithm and we have found the scale factor with the same method as previously (EPnP) by triangulating the points of the even images of the left camera with the odd images of the right camera. The result is shown in blue in Figure 14. As we can observe, the trajectory obtained becomes totally false because of this non-synchronization. This demonstrates the benefit of using our approach in the case of a non-synchronized multi-camera system.

For the second experiment, we have used a very well-known state of the art method, the LIBVISO2 algorithm [14]. We have proceeded in the same way as before, that is to say by using the synchronized pairs and then introducing an offset of one image between the left and right camera to introduce a non-synchronization. Results are shown in Figure 15. The green curve is the ground truth trajectory. The red curve is the result of our method. The black curve is the result using a synchronized stereoscopic system. The blue curve is the result for the non-synchronized case. As we can once again observe, the trajectory

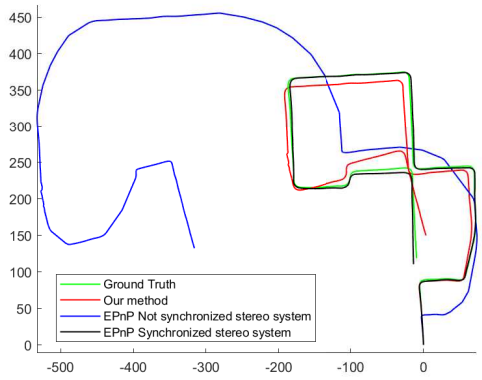


Fig. 14 Comparison of our method with the EPnP algorithm to recover the scale with a stereoscopic system.

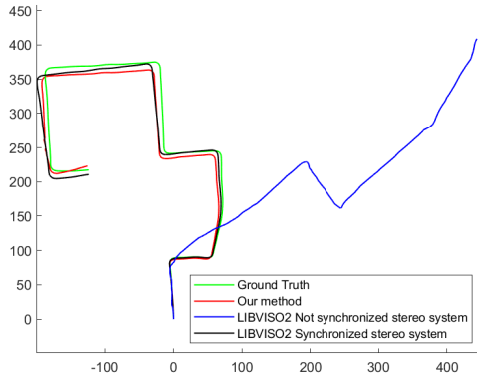


Fig. 15 Comparison of our method with the LIBVISO2 algorithm to recover the scale with a stereoscopic system.

obtained becomes totally false because of this non-synchronization, showing the interest of our method.

5.5 Discussion about the experiments

In Section 5, we have presented the quantitative and qualitative results of our method, and a comparison with stereoscopic algorithms. One might wonder why we did not do more comparisons with existing state-of-the-art methods. As we pointed out in the introduction and in section 2, there is, to the best of our knowledge, no work that has been done on asynchronous systems. Our method can not be compared to other methods for several reasons.

First, it is not possible to compare our algorithm with a monocular method, since in this case the scale factor can not be estimated using only images. It is the main interest of our method, to find the scale factor without having a synchronization between the cameras. We

could obtain it from the ground truth, but in this case it is obvious that our method would be less accurate since we then compare our results to this same ground truth.

Second, it is not fair to compare our algorithm to a synchronized stereoscopic system because in our method, we add a difficulty introduced by the non-synchronization. To solve this problem, we have to make assumptions and our results are logically inferior to these methods. We have nevertheless made this comparison in section 5.4. to show the interest of our method when the stereoscopic system is no longer perfectly synchronized.

Finally, we are aware that the results we are getting are not as good as the best state-of-the-art methods, but this is because we are not dealing with the same problem as mentioned above. Indeed, all methods classified in the "odometry" category of the KITTI dataset are either lidar-based methods or synchronized stereovision-based methods.

6 Conclusion

In this paper, we have proposed a new algorithm for estimating the 3D structure and the motion from a network of asynchronous cameras. As this kind of sensor is increasingly integrated into vehicles for simple tasks such as parking assistance, our goal was to demonstrate its potential use for future higher level applications such as 3D reconstruction around the vehicle. By stating simple hypothesis, such as the linearity of the motion between consecutive views of the same camera and overlapping between adjacent views, our method allows a simple and fast estimation of the relative pose of the cameras including the absolute scale. A bundle adjustment dedicated to the 3D points and scale optimization has also been proposed to improve the results of the initial estimate on a sliding window. All the experiments carried out on the KITTI dataset as well as using our own system have demonstrated qualitatively and quantitatively the validity of our approach. Our future work concerns the dense reconstruction of the vehicle's surrounding environment and obstacle detection.

References

1. Appia, V., Hariyani, H., Sivasankaran, S., Liu, S., Chitnis, K., Mueller, M., Batur, U., Agarwa, G.: Surround view camera system for adas on ti's tdax socs. Tech. rep., Texas Instrument (2015)
2. Baker, S., Nayar, S.: A theory of single-viewpoint catadioptric image formation. International Journal of Computer Vision (IJCV) **35**(2), 175–196 (1999)
3. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: European Conference on Computer Vision (ECCV), vol. 6314, pp. pp 778–792. Heraklion, Crete, Greece (2010)
4. Caspi, Y., Irani, M.: A step towards sequence-to-sequence alignment. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662), vol. 2, pp. 682–689 vol.2 (2000)
5. Clipp, B., Kim, J.H., Frahm, J.M., Pollefeys, M., Hartley, R.: Robust 6dof motion estimation for non-overlapping, multi-camera systems. In: 2008 IEEE Workshop on Applications of Computer Vision, pp. 1–8 (2008)
6. Comport, A., Malis, E., Rives, P.: Real-time quadrifocal visual odometry. International Journal of Robotics Research (IJRR) **29**(2-3), 245–266 (2010)
7. Dhome, M., Mennillo, L., Royer, E., Mondot, F., Mousain, J.: Multibody reconstruction of the dynamic scene surrounding a vehicle using a wide baseline and multifocal stereo system. In: Workshop on Planning, Perception and Navigation for Intelligent Vehicles (satellite event of IROS'17). Vancouver, Canada (2017). URL https://hal-clermont-univ.archives-ouvertes.fr/hal-01657751
8. Dooley, D., McGinley, B., Hughes, C., Kilmartin, L., Jones, E., Glavin, M.: A blind-zone detection method using a rear-mounted fisheye camera with combination of vehicle detection methods. IEEE Transactions on Intelligent Transportation Systems **17**(1), 264–278 (2016)

9. Engels, C., Stewénius, H., Nistér, D.: Bundle adjustment rules. In: In Photogrammetric Computer Vision (2006)
10. Fraundorfer, F., Scaramuzza, D., Pollefeys, M.: A constricted bundle adjustment parameterization for relative scale estimation in visual odometry. In: IEEE International Conference on Robotics and Automation (2010)
11. Furgale, P.T., Schwesinger, U., Rufli, M., Derendarz, W., Grimmett, H., Mühlfellner, P., Wonneberger, S., Timpner, J., Rottmann, S., Li, B., Schmidt, B., Nguyen, T., Cardarelli, E., Cattani, S., Bruning, S., Horstmann, S., Stellmacher, M., Mielenz, H., Köser, K., Beermann, M., Hane, C., Heng, L., Lee, G.H., Fraundorfer, F., Iser, R., Triebel, R., Posner, I., Newman, P., Wolf, L.C., Pollefeys, M., Brosig, S., Effertz, J., Pradalier, C., Siegwart, R.: Toward automated driving in cities using close-to-market sensors: An overview of the v-charge project. In: 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast City, Australia, June 23-26, 2013, pp. 809–816 (2013)
12. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and PatternRecognition (CVPR) (2012)
14. Geiger, A., Ziegler, J., Stiller, C.: Stereoscan: Dense 3d reconstruction in real-time. In: Intelligent Vehicles Symposium (IV) (2011)
15. Häne, C., Heng, L., Lee, G.H., Fraundorfer, F., Furgale, P., Sattler, T., Pollefeys, M.: 3d visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection. Image Vision Comput. **68**, 14–27 (2017)
16. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, second edition edn. Cambridge University Press (2004)
17. Heng, L., Li, B., Pollefeys, M.: Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1793–1800 (2013)
18. Kim, J.H., Hartley, R., Frahm, J.M., Pollefeys, M.: Visual odometry for non-overlapping views using second-order cone programming. In: Y. Yagi, S.B. Kang, I.S. Kweon, H. Zha (eds.) Computer Vision – ACCV 2007, pp. 353–362. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
19. Lébraly, P., Royer, E., Ait-Aider, O., Deymier, C., Dhome, M.: Fast calibration of embedded non-overlapping cameras. In: IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011, pp. 221–227 (2011)
20. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o(n) solution to the pnp problem. International Journal of Computer Vision (IJCV) **81**(2), 155–166 (2009)
21. Levenberg, K.: A method for the solution of certain problems in least squares. Quarterly of Applied Mathematics **2**, 164–168 (1944)
22. Lüke, S., Fochler, O., Schaller, T., Regensburger, U.: Traffic jam assistance and automation. In Handbook of Driver Assistance Systems pp. 1287–1302 (2016)
23. Mathias, M., Timofte, R., Benenson, R., Van Gool, L.: Traffic sign recognition—how far are we from the solution? In: International Joint Conference on Neural Networks (IJCNN), pp. 1–8. Dallas, TX, USA (2013)
24. McCall, J., Trivedi, M.: Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. IEEE Transactions on Intelligent Transportation Systems **7**(1), 20–37 (2006)
25. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Real time localization and 3d reconstruction. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 1, pp. 363–370. IEEE (2006)
26. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time structure from motion using local bundle adjustment. Image Vision Comput. **27**(8), 1178–1193 (2009)
27. Nguyen, T., Lhuillier, M.: Self-calibration of omnidirectional multi-cameras including synchronization and rolling shutter. Computer Vision and Image Understanding **162**, 166–184 (2017)
28. Nister, D.: An efficient solution to the five-point relative pose problem. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **26**, 756–770 (2004)
29. Pless, R.: Using many cameras as one. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA, pp. 587–593 (2003)
30. Pollefeys, M., Nikolic, J., Kneip, L., Kazik, T., Siegwart, R.: Real-time 6d stereo visual odometry with non-overlapping fields of view. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), vol. 00, pp. 1529–1536 (2012)
31. Pooley, D.W., Brooks, M.J., van den Hengel, A.J., Chojnacki, W.: A voting scheme for estimating the synchrony of moving-camera videos. In: Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429), vol. 1, pp. I–413–16 vol.1 (2003)

32. Ramirez, A., Ohn-Bar, E., Trivedi, M.M.: Panoramic stitching for driver assistance and applications to motion saliency-based risk analysis. In: 16th International IEEE Conference on Intelligent Transportation Systems, ITSC 2013, The Hague, The Netherlands, October 6-9, 2013, pp. 597–601 (2013)
33. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: European Conference on Computer Vision (ECCV), pp. pp 430–443. Graz, Austria (2006)
34. Scaramuzza, D., Fraundorfer, F., Siegwart, R.: Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In: 2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009, pp. 4293–4299 (2009)
35. Scaramuzza, D., Fraundorfer, F., Siegwart, R.: Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 4293–4299. IEEE (2009)
36. Schönbein, M., Kitt, B., Lauer, M.: Environmental perception for intelligent vehicles using catadioptric stereo vision systems. In: ECMR, pp. 189–194 (2011)
37. Sons, M., Lauer, M., Keller, C., Stiller, C.: Mapping and localization using surround view. In: IEEE Intelligent Vehicles Symposium (IV), pp. 1158–1163. Los Angeles, USA (2017)
38. Svedman, M.: 3-d structure from stereo vision using unsynchronized cameras. In: Masters thesis, Royal Institute of Technology (KTH (2005)
39. Timofte, R., Zimmermann, K., Van Gool, L.: Multi-view traffic sign detection, recognition, and 3d localisation. Machine Vision and Applications (MVA) **25**(3), 633–647 (2014)
40. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment a modern synthesis. In: Vision algorithms: theory and practice, pp. 298–372. Springer (2000)
41. Wang, Y., Kneip, L.: On scale initialization in non-overlapping multi-perspective visual odometry. In: Computer Vision Systems - 11th International Conference, ICVS 2017, Shenzhen, China, July 10-13, 2017, Revised Selected Papers, pp. 144–157 (2017)
42. Wolf, L., Zomet, A.: Sequence-to-sequence self calibration. In: Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part II, pp. 370–382 (2002)
43. Zhang, B., Appia, V., Pekkucuksen, I., Liu, Y., Umit Batur, A., Shastry, P., Liu, S., Sivasankaran, S., Chitnis, K.: A surround view camera solution for embedded systems. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 662–667. Columbus, USA (2014)

Biography

Rawia Mhiri received her engineering diploma from the Ecole Nationale D'ingégineurs de Sfax (ENIS) in 2011. In 2015, she received her PhD degree from the National Institute of Applied Science Rouen (INSA Rouen). She is now Data Science Technical Lead at Adok.

Safa Ouerghi received her engineering diploma from the Ecole Nationale des Ingénieurs de Monastir (ENIM) in 2010, and her master diploma from the Ecole Supérieure des Communications de Tunis (Sup'Com) in 2012. Her research interests center around vision-based robotic systems and embedded vision applications. Her current research is being focused on the localization and the dynamic security of people flows, objects and information about industrial sites.

Rémi Boutteau received his engineering diploma from the Ecole des Mines de Douai and his MSc degree in computer science and engineering from the University of Science and Technology of Lille (USTL) in 2006. In 2010, he received his PhD degree from the University of Rouen for studies related to computer vision, panoramic vision obtained by catadioptric sensors, and 3D reconstruction algorithms dedicated to omnidirectional vision. After his PhD, he has joined the ESIGELEC engineering school as a lecturer in embedded systems, and the "Instrumentation, Computer Sciences and Systems" research team in the IRSEEM Laboratory. His research interests include computer vision, structure from motion, visual odometry and omnidirectional vision dedicated to autonomous vehicles.

Pascal Vasseur received the MS degree in system control from the Université de Technologie de Compiègne, France, in 1995 and the PhD degree in automatic control from the Université de Picardie Jules Verne, France, in 1998. He was an associate professor at the Université de Picardie Jules Verne in Amiens between 1999 and 2010. He is now full professor at the Normandie Université - Université de Rouen and is a member of the LITIS laboratory. His research interests are computer vision and its applications to intelligent transportation, mobile and aerial robots.

Stéphane Mousset graduated from the Ecole Normale Supérieure of Cachan and received the Ph.D. degree in computer science from the University of Rouen, Rouen, France, in 1997. His Ph.D. dissertation focused on the estimation of axial motion using a stereo vision system. He is currently an Assistant Professor at the Laboratory of Perception, Systems, Information (PSI), National Institute of Applied Sciences (INSA), University of Rouen. He also teaches at the Technology University Institute of Rouen. His research interests include stereo vision analysis, road application, driving assistance systems, and sensors.

Abdelaziz Bensrhair graduated with an MSc in electrical engineering (1989) and a PhD degree in computer science (1992) at the University of Rouen, France. From 1992 to 1999, he was an assistant professor in the Physic and Instrumentation Department, University of Rouen. He is currently a professor in the Computer Science, Information Processing, and Systems Laboratory (LITIS) of the National Institute of Applied Science Rouen (INSA Rouen).