Autonomous Robots (2019) 43:21–35
https://doi.org/10.1007/s10514-018-9698-5

2D–3D synchronous/asynchronous camera fusion for visual odometry

Danda Pani Paudel[1] · **Cédric Demonceaux**[2] · **Adlane Habed**[3] · **Pascal Vasseur**[4]

Received: 2 September 2016 / Accepted: 13 January 2018 / Published online: 1 February 2018

Abstract
We propose a robust and direct 2D–3D registration method for camera synchronization. Once the cameras are synchronized—or for synchronous setups—we also propose a visual odometry framework that benefits from both 2D and 3D acquisitions. Our method does not require a precise set of 2D-to-3D correspondences, handles occlusions and works when the scene is only partially known. It is carried out through a 2D–3D based initial motion estimation followed by a constrained nonlinear optimization for motion refinement. The problems of occlusion and that of missing scene parts are handled by comparing the image-based reconstruction and 3D sensor measurements. The results of our experiments demonstrate that the proposed framework allows to obtain a good initial motion estimate and a significant improvement through refinement.

Keywords Asynchronous cameras · 2D–3D registration · Structure-from-Motion · Visual Odometry

1 Introduction

The problem of accurately localizing cameras is of prime importance in many application involving visual Simultaneously Localization and Mapping (vSLAM). An accurate environment map is generally required for an accurate localization. In turn, building an accurate environment map is not possible without an accurate localization, hence, making it a paradoxical "chicken and egg" problem.

With the ongoing surge in affordable high quality 3D and 2D capture technologies, many mobile robots are, or can easily be, equipped with either or both vision modalities (Holz et al. 2008; Weingarten et al. 2004; Taguchi et al. 2013; Trevor et al. 2012; Bok et al. 2011). We refer to 3D cameras/sensors for any camera that can provide 3D data of the scene directly. For our experiments, we used depth camera, sparse wide angle Lidar sensor, dense narrow angle laser scanner, sparse line scanner laser sensor, and camera-projector scanning setup. As far as 3D sensors are concerned, the Iterative Closest Point (ICP) algorithm (or one of its variants), applied on neighboring 3D point cloud measurements, is overwhelmingly used for robot localization. However, in the case of abrupt or long run displacements, localization based on 3D information alone is difficult mainly because of computational cost and handling degraded (extruded, flat) environments (typical to ICP), and unreliable 3D feature descriptors.

When a robot is equipped with both 3D and 2D sensors, generally 2D images are used to estimate the motion of the cameras (visual odometry) whereas the mapping is obtained directly from the 3D sensor (Buczko and Willert 2016; Pire et al. 2015; Tardif et al. 2010; Jia et al. 2016). Indeed, the emergence of reliable 2D image feature descriptors (such as the Scale-Invariant Feature Transform (SIFT)), 2D-to-2D matching, generally supported by Random Sample Consensus (RANSAC), has become more reliable. However, the accuracy of the camera motion estimation from images, on which the robot localization relies, is undermined by the error amplitude of the extracted 2D features. When localization is based on 2D-to-3D correspondences and 2D–2D based refinement, it may suffer from significant error accumulation. The importance of 2D–3D camera fusion for visual odometry in contrast of 2D–2D based motion refinement (Bok et al.

This research has been funded by an International Project NRF-ANR. DrAACaR: ANR-11-ISO3-0003.

✉ Danda Pani Paudel
paudel@vision.ee.ethz.ch

[1] Computer Vision Laboratory, ETH Zurich, Zürich, Switzerland

[2] Le2i, VIBOT ERL CNRS 6000, Université Bourgogne Franche Comté, Le Creusot, France

[3] ICube UMR 7357, CNRS, University of Strasbourg, Strasbourg, France

[4] LITIS EA 4108, University of Rouen, Rouen, France

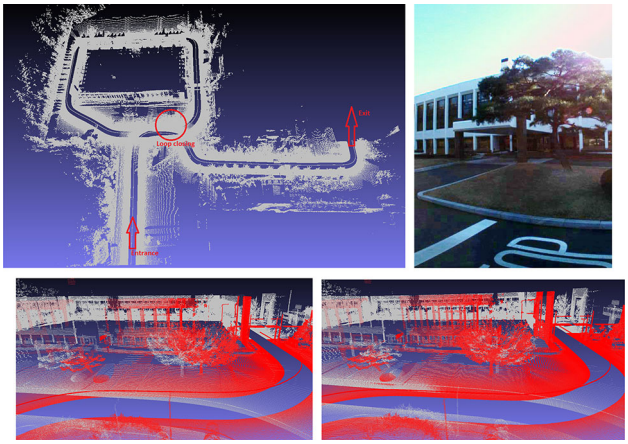


Fig. 1 An example of error accumulation around a loop: Map built by a Laser-Camera system around a large structure (top-left). Image taken at a loop closing point with only one tree at the corner (top-right). Map built before (red) and after (white) the visit around the loop using 2D–2D based refinement (Bok et al. 2011) (bottom-left). Refined map obtained using our method (bottom-right): the scan of the same tree come significantly closer after refinement (Color figure online)

2011), leading to a large error accumulation around a loop, is shown in Fig. 1.

This error is usually minimized by a loop closing technique as described in Williams et al. (2009). However, in particular when robots travel long distances, loop closing is not always possible and may not adequately compensate for error accumulation thus leaving visible artifacts in the map. Performing small and frequent loops are recommended as to keep the accumulated error under control. In practice, making such small loops while building large maps is undoubtedly a burden for the task at hand and often impossible. Though incorporating information from extra sensors such as GPS has been proposed (Bok et al. 2011; Lhuillier 2012), it is often argued that such information is neither accurate nor reliable enough.

Nowadays 3D sensors are providing increasingly high quality and accurate 3D measurements. Therefore, it has now become quite appealing and desirable to jointly benefit from the data acquired from both 2D and 3D modalities to achieve a better localization and/or motion estimation of the cameras at hand. Doing so accurately comes with its fair share of difficulties and challenges. Indeed, 2D and 3D camera setups generally require a full calibration of the system including 2D camera pose with respect to 3D measurements, i.e. extrinsic calibration, and synchronous acquisitions. Maintaining such a calibrated setup is both tedious and difficult due to possible changes in the camera pose parameters and the dedicated hardware required for synchronization. Note that changes in camera pose and/or the presence of synchronization delays, in particular in the case of fast moving systems, may result in large accumulated errors in the long run. Under such circumstances, or when the 3D and 2D captures are asynchronous, the 2D and 3D acquisitions need to be registered before they can be fused. Therefore, we cast the problem of asynchronous cameras as the problem of inaccurate extrinsic between 2D and 3D cameras. For a calibrated 2D and 3D camera-setups under motion, if their acquisitions are synchronized, the extrinsic between them is still valid. However, for asynchronous acquisitions, the extrinsic are not valid anymore due to the motion of the platform/vehicle during the acquisition time gap. Therefore, when dealing with asynchronous cameras, extrinsic parameters need to be re-calibrated/refined jointly with ego-motion estimation.

While in the asynchronous case 2D–3D correspondences are unknown and need to be established, in the calibrated synchronous case, obtaining accurate 2D-to-3D matches is dependent upon the density of the 3D point cloud. Indeed, on the one hand, not every 3D point has known 2D corresponding points and, on the other hand, corresponding image points may not have the exact corresponding 3D point measurement present in the point cloud. Furthermore, whether the system is synchronous or not, some measurements captured by each modality may not be captured by the other. This mainly occurs because parts of the scene may be occluded by others. In the case of 3D captures, this results in (possibly large) missing parts from the scene. This renders the problem of registering data from both modalities rather challenging and difficult to solve.

In this paper, we propose a method for direct 2D–3D registration when 3D and 2D cameras are asynchronous. Once the asynchronous images are registered with the scene, they can be treated as synchronous acquisitions for which we propose a complete visual odometry framework that combines both 2D and 3D data. The proposed asynchronous 2D–3D registration method demands only a rough knowledge of the pose of only one of the cameras and, apart from 3D scene point coordinates, requires no other knowledge regarding the geometry of the input scene. We assume that point correspondences across images are available but 2D-to-3D correspondences are unknown. To our knowledge, there is no method that makes use of both 2D and 3D information without 2D-to-3D correspondences. Note that methods employing Bundle Adjustment (BA) with known scene (Triggs et al. 2000) and PnP (Hesch and Roumeliotis 2011) require such 2D-to-3D correspondences to be established. In practice, good 2D correspondences between instantaneously captured images can be obtained by using state-of-the-art feature descriptors such as SIFT. The proposed method does not require a precise set of 2D-to-3D correspondences, handles occlusions, and works for partially known scenes. This framework computes the pose by localizing a set of cameras at once with respect to the 3D scene acquired in the previous frame using a minimum of three corresponding points among all the views. Furthermore, a

constrained nonlinear optimization framework is also proposed for pose refinement. The first step of visual odometry uses only the known part of the scene whereas our refinement process uses the constraints that arise from the unknown part as well. The refinement step minimizes the projection errors of 3D points while enforcing the existing relationships between images. Both steps handle the problem of occlusion and that of missing scene parts by confronting the image-based reconstruction and the 3D sensor measurements. They also minimize the effect of data inaccuracies by using an M-estimator based technique. Unlike (Tamaazousti et al. 2011), our method makes no prior assumption regarding the geometry of the scanned scene. The presented method differs from its preliminary works (Paudel et al. 2014a, b) as it introduces both synchronous and asynchronous systems under a common framework. Furthermore, we also provide the results for cascaded asynchronous to synchronous model.

Our paper is organized as follows. Related work is presented in Sect. 2. The notations used in the present paper and the necessary background are introduced in Sect. 3. We formulate the optimization problem to obtain the optimal odometry parameters in Sect. 4. The solution to this problem is presented in the form of an algorithm in the same section. In Sect. 5, experiments with synthetic and four real datasets are presented and discussed. Section 6 concludes our work.

2 Related work

The 2D–3D registration problem is tackled in the literature through direct and indirect approaches. The direct registration methods rely on establishing feature correspondences (such as points, lines, planes, skylines and building bounding boxes) between the images and the 3D scene. The point-based matching methods proposed in Sattler et al. (2011), Knopp et al. (2010) require the 3D scene along with a scale invariant feature descriptor (SIFT) for each point. Correspondences are obtained by matching these feature descriptors to that of image feature points. Establishing reliable correspondences may be undermined by the absence of such descriptors in the provided scene points as well as by the variability of the illumination conditions during the 2D and 3D acquisitions. Methods relying on higher level features, such as lines (Christy and Horaud 1999), planes (Tamaazousti et al. 2011) and building bounding boxes (Liu and Stamos 2005), are generally suitable for Manhattan World scenes (or the like) and hence applicable only in such environments. Skylines-based methods (Ramalingam et al. 2009) as well as methods relying on a predefined 3D model (Clarkson et al. 2001) are, likewise, of limited applicability.

Indirect methods are performed either by 3D–3D registration or by finding some appropriate registration parameters. Methods based on 3D–3D registration are performed using the (rigid or non-rigid) Iterative Closest Point (ICP) algorithm between the Structure-from-Motion (SfM) induced reconstruction and the known scene. Some alternative methods use probabilistic approaches for 3D–3D registration. For instance, Horaud et al. (2011) uses expectation conditional maximization, Stoyanov et al. (2012) uses normal distributions transforms, and Evangelidis et al. (2014), Eckart et al. (2015) use Gaussian mixture models. Although there exists several other techniques for scaled point clouds registration (Pomerleau et al. 2015), there extension for registering point clouds with unknown reconstruction scale is not straightforward. For instance, this scale ambiguity is handled by an extension of the 4-point congruent sets algorithm in Corsini et al. (2013). On the other hand, registration based on complex parameters, such as mutual information (Viola et al. 1997) and region segmentation (Taneja et al. 2012), are based on single images. Therefore, each camera requires its own initialization and is individually localized independently from the rest of the cameras. Cameras that are localized in this fashion may fail to satisfy the multiview geometric constraints (such as the epipolar constraint in two images). In this context, the proposed registration method belongs to direct registration categorty. Starting from a rough knowledge of camera position, our method performs direct registration between 2D and 3D point sets without requiring the point-to-point correspondences.

Visual odometry is generally carried out by relying on 2D–2D, 3D–3D, or 2D–3D information. 2D–2D based methods typically track features in monocular or stereo images and estimate the motion between them (Chiuso et al. 2000; Nister et al. 2004). Some of these methods improve the localization accuracy by simultaneously processing multiple frames, while using BA for refinement. Some other methods obtain the motion parameters by registering images such that the photometric error between them is minimized (Koch 1993; Comport et al. 2007). For the same purpose, most 3D–3D based methods use ICP or its variants (Besl and McKay 1992; Fitzgibbon 2003; Rusinkiewicz and Levoy 2001) between consecutively acquired point clouds obtained from the 3D camera (Nüchter et al. 2007; Newcombe et al. 2011). However, ICP-based methods are computationally expensive due to the calculation of the nearest neighbors for every point at each iteration. Both of these methods use the information from either camera only and, hence, do not fully exploit all the available information.

Recent works (Tamaazousti et al. 2011; Kerl et al. 2013) propose the use of information provided from both cameras during the process of localization. The work in Tamaazousti et al. (2011) refines the camera pose obtained from Structure-from-Motion (SfM) using an extra constraint of a plane-induced homography via scene planes. This method provides a very good insight for a possibility to improve the camera pose when the partial 3D is known. However, it

uses only the information from planes that are in the scene. The methods presented in Newcombe et al. (2011), Kerl et al. (2013) and Henry et al. (2012) have been tested in indoor environments mainly with a Kinect sensor. Extension of these methods to outdoor environments with possibly different kinds of 3D cameras is not trivial due to various unhandled situations that may arise. Typical issues arising in outdoor scenes and/or different camera setups occur, for example, when 2D and 3D cameras do not share the exact same field of view, when the 3D points are sparse (as opposed to pixel-to-pixel mapping of RGB-D cameras), in the absence of required scene structures, and in the event of low frame rates and/or large displacements of the cameras.

Another work by Zhang et al. (2014) uses both 2D and 3D cameras for outdoor visual odometry using synchronous camera setup. This method takes advantage of multi-frame motion estimation as well feedback model. The final odometry parameters are then refined by BA from iSAM2 (Kaess et al. 2011). Although this method performs well for synchronous setups, it does not address the problem of camera fusion for asynchronous setup. Note that other existing 2D–3D based refinement methods, such as BA and loop closing, are not applicable under these circumstances because they require precise 2D-to-3D correspondences across frames. In this work, the pose refinement is carried our using both 2D and 3D information. Our refinement method successfully uses both 2D and 3D even when precise 2D-to-3D correspondences are not known. In particular to asynchronous cameras, Rawia et al. (2014) uses asynchronous 2D camera rigs for intelligent vehicle application. Ego-motion estimation in Rawia et al. (2014) is carried out for 2D camera rigs, unlike 2D–3D camera setup in this work. Furthermore, Rawia et al. (2014) makes a very strong assumption that the ego-motion is piecewise linear. Such assumption might be valid for intelligent vehicle setups (as originally developed). However, it is not practical in general case (for example, hand-held cameras).

3 Notation and background

The setup consists of a 3D scanner and multiple calibrated cameras as shown in Fig. 2. We refer the 3-space Euclidean transform by a 4×4 matrix $T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$, where R and t are rotation matrix and translation vector respectively. At any given instant, the 3D scanner scans the scene points $X_k, k = 1 \ldots p$ in its coordinate frame O^1. A set of calibrated cameras at $T_i, i = 1 \ldots m$, not necessarily overlapping, capture m images, from which a set of 2D feature points are extracted. Let $x_{ij}^1, j = 1 \ldots n$ represent those feature points in the ith image. $P(T, X)$ is the projection function that maps a point X to its 2D counterpart in the image captured

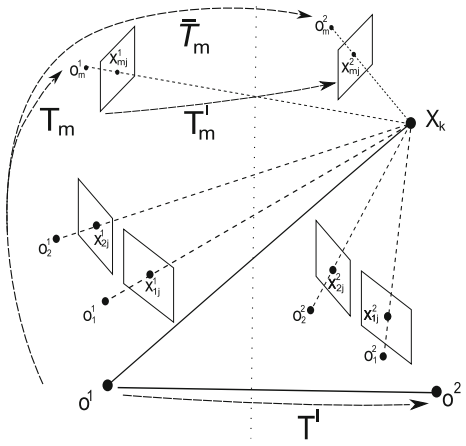


Fig. 2 Ray diagram of the experimental setup

from T. When the system moves by T' to next position, corresponding variables are represented by the same notations with change in superscript. The poses of the second set of cameras with respect to O^1 are expressed as $\bar{T}_i$. The Essential matrix between two views of the same camera in different frames is expressed by $E_i(T')$, with an abuse of notation for simplicity. Although $E_i(T')$ is expressed as the function of T', it is actually the function of T'_i, which are again dependent upon both T' and T_i. For synchronous setups, the transformation matrices are related follows

$$T'_i = T_i T' T_i^{-1}. \tag{1}$$

If x_{ij}^1 and x_{ij}^2, $j = 1 \ldots n$ are corresponding feature points in two consecutive images taken by the ith camera, their 2D-to-3D correspondences are specified by a function ϕ. Let $\phi_i(j)$ be a function that maps each pair of 2D points $x_{ij}^1 \leftrightarrow x_{ij}^2$ to the corresponding 3D point X_k. Every rotation matrix is represented by a 4×1 vector of quaternions, unless mentioned otherwise. Whenever the estimation of rotation is involved, the unit norm of quaternions is assumed to be enforced. Both 3D and 2D points are represented by 3×1 vectors, the latter being the homogeneous representation in the camera coordinate system. The distance between two rotation matrices is measured by computing the spectral norm of their difference. For a matrix A, its spectral norm is denoted as $|||A|||$. Two given up-to-scale translation vectors are compared by measuring the angle between them.

4 2D–3D visual odometry

In this section, we establish the relationships between a set of image pairs and scene points. Using these relationships, we propose an optimization framework whose optimal solution is the required odometry parameters. A complete algorithm for solving this optimization problem is also discussed. The

proposed method deals with both the asynchronous and synchronous cases separately. In the asynchronous case, the camera's extrinsic parameters T_i' are assumed to be unknown. In the synchronous case these parameters are known and fully exploited during the motion estimation process. We also assume that the 2D-to-2D correspondences between image pairs acquired by the same camera are known.

4.1 Problem formulation

The relationship between 2D and 3D points is depicted in the ray diagram given in Fig. 2. The projection error of points on the first set of cameras is given by

$$e^1(T_i, \phi_i(j)) = ||x_{ij}^1 - P(T_i, X_{\phi_i(j)})||^2. \tag{2}$$

Similarly, for the second set of cameras, the projection error is given by

$$e^2(T_i, T', \phi_i(j)) = ||x_{ij}^2 - P(T_i T', X_{\phi_i(j)})||^2. \tag{3}$$

Furthermore, the epipolar constraint that relates the points in two views of different frames can be written as

$$\left(x_{ij}^2\right)^T E_i(T') x_{ij}^1 = 0. \tag{4}$$

While (2) locates the first camera, (3) locates the second camera with respect to the world reference frame while preserving its relationship to the first one. Similarly, (4) localizes the second camera with respect to the first one. Equations (2), (3) and (4) are obviously redundant. However, in the presence of noise in the data and unknown correspondences all constraints must be enforced: satisfying only the non-redundant conditions does not necessarily satisfy all of them. In addition, (4) makes use of the unknown part of the scene as well. Therefore, all three equations will be incorporated in our optimization framework in which (3) is chosen to be the objective (as it includes the pose of both the cameras) while the rest are used as constraints.

Our problem is to localize a set of 2D cameras with known 2D-to-2D ($x_{ij}^1 \leftrightarrow x_{ij}^2$) and unknown 2D–2D-to-3D ($x_{ij}^1 \leftrightarrow x_{ij}^2 \leftrightarrow X_{\phi_i(j)}$) correspondences in the presence of noise. Hence, finding the optimal ϕ_i itself is part of the optimization process. Therefore, the optimization framework can be written as

$$\begin{aligned} \min_{T_i, T', \phi} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} ||x_{ij}^2 - P(T_i T', X_{\phi_i(j)})||^2, \\ \text{s.t.} \quad & ||x_{ij}^1 - P(T_i, X_{\phi_i(j)})||^2 = 0, \\ & (x_{ij}^2)^T E_i(T') x_{ij}^1 = 0. \end{aligned} \tag{5}$$

The optimization problem (5) considers that every image point has its corresponding 3D point in the scene. In practice, there could be extra 2D or missing 3D points resulting in invalid 2D-to-3D correspondences. We address these problems by assigning the weights derived from a scale histogram to each correspondence. Furthermore, we also relax the strict equality of constraints to avoid the infeasibility that would arise due to the noisy data (or the discretization during the image formation process).

If $\widetilde{X}_{ij}$ is the SfM reconstruction in O^1, the relative scale of reconstruction for known 3D-to-3D correspondences $\widetilde{X}_{ij} \leftrightarrow X_{\phi_i(j)}$ is given by $s_i(j) = ||\widetilde{X}_{ij}||/||X_{\phi_i(j)}||$, $j = 1 \ldots m$. Since the reconstructed points from each pair share a common scale, in the ideal case, we have $s_i(j) = c_i$. $\forall j \in 1 \ldots n$, for some constant c_i's. In practice, when the histograms $H_i(u)$, $u = 1 \ldots b$ of these scales are built, they hold the highest number of samples in the bin corresponding to the true scale. If those bins are U_i, then the weights are distributed as follows:

$$w_i(j) = \begin{cases} 1 & s_i(j) \in H(U_i) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Furthermore, the effect of data inaccuracies is reduced by introducing a robust estimation technique. Hence, the optimization problem (5) with robust estimation and histogram-based weighting can be re-written as

$$\begin{aligned} \min_{T_i, T', \phi} \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} w_i(j) \rho\left(||x_{ij}^2 - P(T_i T', X_{\phi_i(j)})||\right), \\ \text{s.t.} \quad & w_i(j) \rho\left(||x_{ij}^1 - P(T_i, X_{\phi_i(j)})||\right) = 0, \\ & \rho\left((x_{ij}^2)^T E_i(T') x_{ij}^1\right) = 0. \end{aligned} \tag{7}$$

where $\rho(x)$ is Tukey bi-weighted potential function. For a threshold ξ, it is defined as

$$\rho(y) = \begin{cases} \frac{y^6}{6} - \frac{\xi^2 y^4}{2} + \frac{\xi^4 y^2}{2} & for\ |y| < \xi \\ \frac{\xi^6}{6} & \text{otherwise} \end{cases} \tag{8}$$

whose influence function is $\psi(y) = y\left(\xi^2 - y^2\right)^2$ for $|y| < \xi$ and 0 otherwise.

Note that any 2D-to-3D correspondence that does not vote for the valid scale is considered to be an outlier. Here, the derived cost depends only upon the known part of the scene whereas the constraint includes the unknown part as well. The optimal odometry parameters are obtained by iteratively solving this optimization problem. Each iteration breaks the problem down into two subproblems: (a) 2D-to-3D registration and (b) Camera pose refinement.

Table 1 Known and estimated parameters

	Input	Estimation
Asynchronous	2D–2D corresp.	T_i and $\bar{T}_i$
Synchronous	2D–2D corresp., T_i	T'

4.2 2D-to-3D registration

The registration step coarsely localizes the cameras with respect to the scene. Here, we discuss the registration methods for asynchronous and synchronous cases as two separate subproblems. In the asynchronous case, finding the 2D-to-3D correspondences required for registration is not trivial. This is done by iterating between camera poses and the correspondence estimation. On the other hand, finding the precise cross-frame correspondences for the synchronous case is not easy either. Cross-frame image-to-scene registration in synchronous acquisition is carried out by using minimal point RANSAC-based pose estimation. The choice of registration methods depends upon the experimental setup. The known input and estimated parameters for two different cases are summarized in the Table 1.

4.2.1 Asynchronous case

The main problem in the asynchronous acquisition is that the poses of the camera with respect the scene are unknown. This makes solving 2D-to-3D correspondence problem very challenging. Since these correspondences are unknown, the reconstruction that can be obtained from images is related to the scene by an unknown scale factor. To avoid the role of this unknown scale, we minimize a cost function which is independent of it, while imposing the epipolar constraint between images. The proposed optimization problem for asynchronous cameras registration is as follows:

$$\min_{T_i,\phi} \sum_{i=1}^{m}\sum_{j=1}^{n} w_i(j)\rho\big(||(x_{ij}^2)^T E_i(T')P(T_i, X_{\phi_i(j)})||\big),$$
$$\text{s.t. } w_i(j)\rho\big(||x_{ij}^1 - P(T_i, X_{\phi_i(j)})||\big) = 0. \tag{9}$$

The initial estimate of T_i' is obtained using the SfM-based relative pose estimation method (Nister 2004). Note that T' is the motion between the 3D cameras, whereas T_i' are the same for 2D cameras. In this case, we choose ϕ such that it maps every pair of image points to a 3D point that respects the constraint while minimizing the cost. The constraint violation is penalized by a simple but effective static penalty function as discussed in Smith and Coit (1995). Therefore,

$$\phi_i(j) = \underset{k\in\{1,\ldots,p\}}{\arg\min}\ ||x_{ij}^1 - P(R_i, t_i, X_k)|| + ||\left(x_{ij}^2\right)^T E_i(T')P(T_i, X_k)||. \tag{10}$$

Hence, the optimal poses of the first set of cameras are obtained, for each camera i separately, by solving

$$\underset{T_i}{\arg\min} \sum_{j=1}^{n} w_i(j)\rho\big(||(x_{ij}^2)^T E_i(T')P(T_i, X_{\phi_i(j)}^1)||\big),$$
$$\text{s.t. } w_i(j)\rho\big(||x_{ij}^1 - P(T_i, X_{\phi_i(j)})||\big) = 0. \tag{11}$$

This is a constrained nonlinear optimization problem on the quaternion parameters whose local optimal solution can be obtained by the iteratively re-weighted least-squares (ILRS) technique. In fact, depending upon one's choice, it can also be solved linearly on R and t using singular value decomposition. However, the linear solution does not constrain R to be a rotation matrix. Therefore, the obtained solution needs to be enforced as a rotation matrix before extracting the quaternion parameters.

For each pair of images, the scale of the reconstruction is finally estimated by averaging the scales of inliers as follows

$$\mu_i = \frac{\sum_{j=1}^{n} w_i(j)s_i(j)}{\sum_{j=1}^{n} w_i(j)}, \quad i = 1\ldots n. \tag{12}$$

Finally, the absolute poses of the second set of cameras in O^1 can be obtained through

$$\bar{T}_i = \begin{pmatrix} R_i' & \mu_i t_i' \\ 0 & 1 \end{pmatrix} T_i. \tag{13}$$

Recall that R_i' and t_i' are the rotation and translation components of T_i'. Once the cameras are fully registered, they can be thought as synchronized ones. This is because the second set of cameras can be localized in the first coordinate frame. Henceforth, we consider two cases: (1) Asynchronous model assumes that the cameras are not yet synchronized; (2) Cascaded asynchronous-to-synchronous assumes that the asynchronous cameras are synchronized via 2D–3D registration.

Under the assumption that T_i is known with scale, 3D points from reconstruction can be directly associated to points from the 3D sensor. In fact, once these points are aligned by T_i, they differ only by a scale factor $s_i(j)$. First, we obtain $s_i(j)$ by taking the ratio of their norms. Then, absolute scale for each camera is obtained by (12). On the other hand, one can obtain T_i with scale, due to the constraint imposed in (9). This constraint can also be thought as solving the perspective n-point problem. For a correct set of $\phi_i(j)$, T_i can be estimated with the correct scale. Therefore, ego-motion of each image can be estimated by using (13). If the ego-motion of the 3D sensor is required, it can be obtained by using (1).

4.2.2 Synchronous case

It is trivial to find the 2D-to-3D correspondences $X_k \leftrightarrow P(T_m, X_k)$ in one frame. However, cross-frame correspondences are required in order to estimate T'. Such correspondences can be obtained by matching the 2D feature points between images. Note that most $P(T_m, X_k)$, when considered as feature points, are unlikely to result in reliable feature descriptors for matching. Therefore, we extract a separate set of 2D feature points to obtain better 2D–2D correspondences $x_{ij}^1 \leftrightarrow x_{ij}^2$. Methods based on relative pose require at least 5 such correspondences to compute the motion with an unknown scale. On the other hand, if 2D-to-3D correspondences $x_{ij}^2 \leftrightarrow X_k$ can be found, it would require only 3 points to estimate the motion including the scale. In order to benefit from this, the required 2D-to-3D correspondences are computed for each image which is established by the mapping function $\phi_i(j)$ computed as

$$\phi_i(j) = \underset{k\in\{1,\ldots,p\}}{\arg\min} \; ||x_{ij}^1 - P(T_i, X_k)||, \; j = 1 \ldots n. \tag{14}$$

It is important to notice that the correspondences obtained in this manner are not perfect. We make a strong consideration of this restriction while refining the estimated motion. The search required to minimize (14) can be performed using a KD-tree like structure where the projections of all 3D points build one tree in each image. The detected feature points traverse these trees in search for the best possible match. Once the required correspondences are obtained, the set of cameras in the second frame can be localized with respect to previously acquired 3D scene using the method presented in Nister (2004). The advantage of using this method is that it requires a minimum of 3 correspondences among all the views and does not require a complex scene as demanded by ICP or SfM. For example, even a planar scene with sufficient texture can be processed. For low frame rates and/or large displacements, feature matching methods still work better than tracking them. Since only 3 correspondences are needed, finding them from already matched 2D–2D to sparse 3D is very much achievable in practice.

4.3 Camera pose refinement

Recall that in both asynchronous and synchronous cases the final result is the registration of next frame images to the previous scene. In fact, the obtained registration parameters are the absolute poses of the cameras. However, in practice, the motion obtained in this manner is not very accurate. In this step, we refine these coarse motion/registration parameters while making use of scene information. The refinement process optimizes the motion parameters such that the SfM reconstruction is closest to the known scene. During this process, the asynchronous setups are refined by directly solving (7) for the known correspondence function ϕ. The correspondences required in this step are obtained directly from the registration process. However, the synchronous setups are refined by solving the following optimization problem:

$$\begin{aligned} &\min_{T'} \; \sum_{i=1}^{m}\sum_{j=1}^{n} w_i(j)\rho\big(||x_{ij}^2 - P(T_i T', X_{\phi_i(j)})||\big), \\ &\text{s.t.} \; \rho\big((x_{ij}^2)^T E_i(T') x_{ij}^1\big) = 0. \end{aligned} \tag{15}$$

Note that the refinement process uses all the cameras simultaneously to refine T', unlike in (11) of the asynchronous case. This is again a constrained nonlinear optimization problem that can be solved by ILRS technique. Each iteration of IRLS uses the interior-point method to solve the constrained nonlinear least-squares problem.

4.4 The algorithm

Starting from known 2D-to-2D correspondences, the algorithm iteratively estimates the odometry parameters mentioned in Table 1. Every iteration reduces the cost function (7) in two steps while satisfying its constraints. Here, we present two different algorithms for asynchronous and synchronous cases separately.

Algorithm 1 Asynchronous case

For known initial guess on T_i and T_i' obtained from relative pose estimation, refine them through the following two steps:

1. **Camera alignment**: iteratively align the cameras to scene until convergence,
   (a) estimate the relative pose using 2D-to-2D correspondences;
   (b) compute 2D-to-3D correspondences using (10);
   (c) build multiple scale histogram $H_i(u)$ and compute weights $w_i(j)$, $j = 1 \ldots n$;
   (d) update the pose of the first set of cameras using (11).
2. **Simultaneous pose refinement**: starting from the results obtained in the "Camera alignment" step, refine poses of both sets of cameras by solving (7) for known ϕ.

Obtain real scale μ_i and compute the absolute pose using (13).

Note that the cascaded asynchronous-to-synchronous model uses (15) instead of (7) in the refinement step of Asynchronous algorithm.

4.5 Normalization and pose recovery

For the sake of numerical stability, the 3D scene points are normalized such that the distance between the scene's centroid to the first camera is approximately equal to 1. If the

Algorithm 2 Synchronous case

1. **2D–3D registration**: for known extrinsics $T_i, i = 1 \dots m$, iterate over the following steps until convergence:
   For each Camera $i = 1 \dots m$
   (a) compute $P(T_i, X_k), k = 1 \dots p$ and build a KD-tree;
   (b) find 2D-to-3D correspondences maps $\phi_i(j), j = 1 \dots n$ using (14).

   Using all Cameras: perform 2D–3D-based RANSAC and estimate $T'_{i,0}$ using Nister (2004).
2. **2D–2D-to-3D based refinement**: starting from $T'_{i,0}$, iterate until convergence,
   (a) Reconstruct the scene $\widetilde{X}_{ij}, j = 1 \dots n$ and compute scales $s_i(j)$ for each point;
   (b) Build a combined scale histogram $H(u), u = 1 \dots b$ for all cameras;
   (c) Compute weights $w_i(j), j = 1 \dots n$ using $H(u)$;
   (d) Update the pose by optimizing (15) for known $\phi_i(j)$ obtained from 2D–3D registration.

initial estimate of T_i's are $\{R_{i,0}, t_{i,0}\}$, such normalization corresponds to $\widehat{X^i} = (R_{i,0}X + t_{i,0})/||t_{0,i}||, \ i = 1 \dots m$. After this transformation, $R_{i,0}$ and $t_{i,0}$ simplify to $I_{3\times3}$ and $0_{3\times1}$ respectively. We also normalize the data during the robust estimation i.e. y in (8) is scaled with twice of its median value and ξ is set to 1 whenever it is used. The iterations are terminated when the improvement of the pose between two consecutive iterations $k-1$ and k of both cameras becomes insignificant. The improvements on the rotational R and translational t components are computed using

$$e_R = |||R_k - R_{k-1}||| \text{ and } e_t = cos^{-1}\left(\frac{t_k^T t_{k-1}}{||t_k||||t_{k-1}||}\right). \quad (16)$$

Improvements on R' and t' are also computed similarly. The algorithm terminates when $e_R < \zeta_1, e_{R'} < \zeta_1, e_t < \zeta_2$, and $e_{t'} < \zeta_2$ for some given thresholds ζ_1 and ζ_2.

4.6 Discussion

The problem addressed here is similar to that of the scaled variant ICP as in Zhao et al. (2005). The solution to (9) provides the scaled-ICP-like registration of image-sets in a direct manner. However unlike (Zhao et al. 2005), where the 3D-to-3D correspondences are searched, we established 2D-to-3D direct correspondences using (10). Once the correspondences are found, the Eq. (11) refines the registration parameters, in a very usual ICP-based methods. Algorithm 1 describes the steps for Asynchronous case. Here, step 1 (Camera alignment) only aligns the image-sets with respect the 3D scene, whereas step 2 (Simultaneous pose refinement) refines the pose using coarse alignment obtained from step 1.

Regarding the choice of T_i, once the essential matrix is fixed, for a dense 3D scene, one can always find a 3D point that lies on the ray back-projected from the image point. However, for any T_i, the 3D point lying on the ray does not share a common scale with rest of the others. Thanks to the scale histogram, a 3D point belonging to common scale with rather some error (due to inaccurate current T_i estimates) is selected. Now, since the 3D point is not error free, its projection on the image does not necessarily satisfy either the cost or the constraint. Furthermore, due to such trade-off between scale and the point on back-projection ray, satisfying the constraint does not necessarily satisfy the cost, or vice versa.

5 Experiments

We tested our methods using both synthetic and real datasets. Our results with synthetic data were compared against those of ICP with SfM. For real data, experiments with four different datasets captured under different setups were performed. In all the cases, the constrained nonlinear least-squares optimization problem was solved by using MATLAB-R2012a Optimization Toolbox with interior-point method. The computational time for the experiments varies upon various factors, mainly on the number of 2D points. With the increase of 2D points, the number of constraints increases, and hence the computational time. A typical real data experiment of 314 2D points takes 1.76 s for synchronous case and 4.05 s for asynchronous case. Note that the code was implemented in MATLAB and not optimized. All experiments were carried out on a 8GB RAM Pentium i7/3.40GHz. [1]

5.1 Simulations

We generated a set of 800 random 3D points scattered on the surface of four faces of a $[-10\ 10]^3$ cube. The cameras were placed about 20 ± 2 units away from the origin with randomly generated rotations while roughly looking towards the centroid of the scene. All scene points were projected onto 256×256 images with zero-skew, 100 pix. focal length and an image-centered principal point. The 2D data were obtained by adding various levels of zero-mean Gaussian noise to the pixel coordinates. 400 out of 800 projected points were randomly selected and used to localize the second camera with respect to the first one using classical SfM (Nister 2004). During this process, half of the points are rejected to minimize the effect of outliers thus leading to the reconstruction of only 200 points. The same data were used in our method to perform the registration and the refinement. We ran 100 tests for each noise level of standard deviation from

[1] Compilation of few results as a supplementary video can be found at: https://www.youtube.com/watch?v=iPYOgBAMUZc&feature=youtu.be

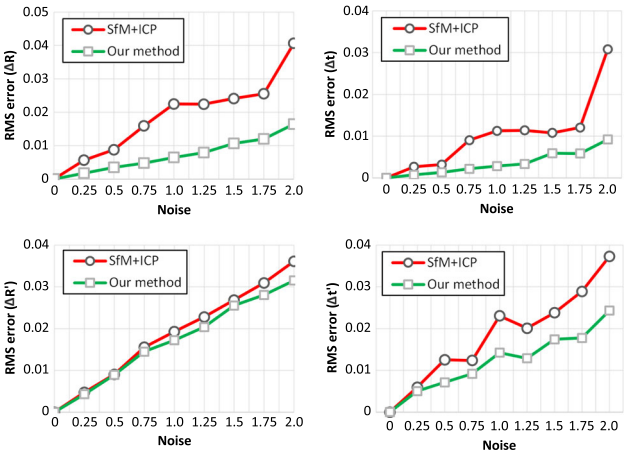


Fig. 3 SfM+ICP vs. Our method with noise; ΔR (left-top), Δt (right-top), $\Delta R'$ (left-bottom), and $\Delta t'$ (right-bottom)

0 to 2.0 with a 0.25 step. The simulation results are presented for the two-view case only.

The roughly known R was generated by introducing an error of $[0.05\ 0.075]^c$ in roll, pitch and yaw each. We introduced these relatively small errors in R to observe the improvement when the iterative scheme converges. Similarly, a small error of $\pm\,5\%$ was introduced in each translation axis. Nevertheless, these errors are very significant since the scene is relatively far from the cameras. The histogram was built with auto adjustable 10 bins after discarding the scales of less than 0.1 and greater than twice its median. First, we obtained the best possible R, t, R', and t' using classical SfM (Nister 2004) and ICP (Martin and Jakob 2012). As ICP cannot be performed without the knowledge of relative scale, the extra information of scale is recovered with the assumption of the image-based reconstruction being spread all over the provided 3D scene. Note that, our method does not require this extra information of scale. To analyze the improvement on camera pose, we computed the deviation of these results from their ground truth values. The errors ΔR, Δt, $\Delta R'$, and $\Delta t'$ correspond to the residuals computed as in (16). Figure 3 shows the Root-Mean Square (RMS) plots of the computed errors for various levels of noise. It can be seen that our method performs significantly better than SfM with ICP even when the ICP is favored with extra information of scale.

5.2 Real data

Four benchmark and one in-house real datasets were used to test the proposed algorithms. Three out of these five datasets were acquired asynchronously and the other two synchronously. Each of these datasets were acquired by very different setups as discussed below. The results obtained were compared against the ground truth (whenever available) or the known desired output. Required 2D-to-2D correspondences for all the experiments were obtained by the SURF descriptor based matching.

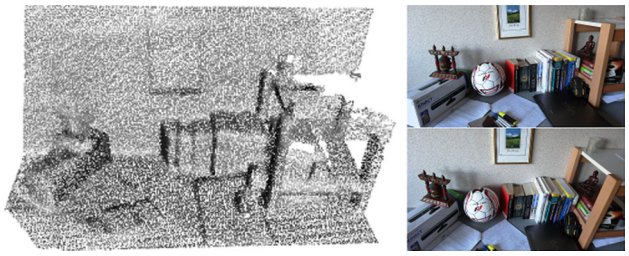


Fig. 4 Left: Kinect 3D scene; Right: image pair

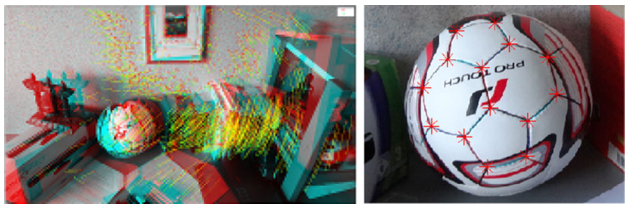


Fig. 5 Left: Correspondences; Right: feature points

5.2.1 Asynchronous case

Scene and images were captured by two different devices. The first dataset was captured by a Kinect sensor and a separate 2D camera. The second dataset consists of two different scenes scanned by a laser-scanner and multiple images captured by a camera. The inputs to our method were the 2D-to-2D correspondences across images, rough absolute pose of the first camera T_1, and the relative pose between images. Our method outputs the corrected relative and absolute poses of all the cameras. Results for the second dataset were compared against the provided ground-truth values. However, the results of the first dataset were compared against the desired reconstruction.

Kinect Dataset For the first experiment with real data, we built the prior 3D scene by registering multiple frames acquired from a 3D sensor (Kinect). This scene was then down-sampled to about 50,000 points as shown in Fig. 4 (left). After the 3D scene is acquired, a standard-sized football was placed in the same scene and two 1080×1920 images were captured by a moving camera. These images and their 1198 correspondences are shown in Figs. 4 and 5. 14 manually selected points from the corners of the Truncated Icosahedron (TI) (Fig. 5 (right)) were retained for assessing the quality of the reconstruction. To overcome the problem of initialization, the first views of both 2D and 3D cameras are captured approximately from the same location while facing towards the same part of the scene.

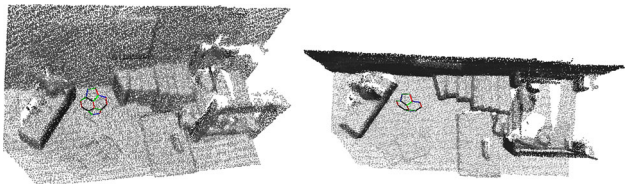


Fig. 6 Two views of the 3D scene with TI

Table 2 Geometric parameters

	LS	AP (cm)	AH	A-HP	A-HH	CS (cm)
SfM	0.201	4.267	2.008	6.195	140.19	76.25
Our method	0.117	2.943	0.863	3.342	139.20	73.10

The final metric reconstruction of the scene is upgraded to Euclidean for the measured length of polygon sides equal to 4.5 cm. Although our method outputs the scale factor for the upgrade, we used the same scale factor for both SfM and our method to provide the comparison on a common ground. Reconstructed TI from two views is placed in the given 3D scene and shown in Fig. 6. We have approximated the circumference of the football by fitting a sphere passing through the vertices of the reconstructed TI. For a quantitative analysis, the following geometric parameters of reconstructed TI are computed: (i) LS: RMS error of the length of sides. (ii) AH: RMS error of the internal angles of hexagons. (iii) AP: RMS error of the internal angles of pentagons. (iv) A-HP: RMS error of Dihedral angles between hexagons and the pentagons. (v) A-HH: Dihedral angle between two hexagons (expected: 138.19). (vi) CS: Circumference of the sphere (expected: 68–70 cm). Table 2 compares these parameters against FIFA's standard. This is an example of 2D-to-3D data fusion where the reconstruction from two views is added to the 3D scene. This example also demonstrates the handling of occlusion problem because of the football placed in the scene after the 3D acquisition. Furthermore, even when the 3D data is not very accurate, as in this case, it shows that our method still benefits from the scene information.

EPFL dataset We also tested our method with the public datasets Fountain-P11 and Herz-Jesu-K7 (Fig. 7 from http://cvlabwww.epfl.ch/~strecha). These datasets consist, respectively, of 11 and 7 images of size 3072×2048 along with ground truth partial 3D point clouds of the scenes. To validate the ground truth, the texture was mapped on the scene by back-projecting images using their ground truth projection matrices. Although the images were taken from different viewpoints, they share a common field of view in the 3D space. If all the images are not aligned correctly with respect to the 3D model, the mapped texture in 3D leaves many artifacts. Therefore, a high quality 3D texture mapping demonstrates the correct registration of asynchronized images and reconstructed 3D scene. Note that our setup

Fig. 7 Left: Fountain-P11; Right: Herz-Jesu-K7

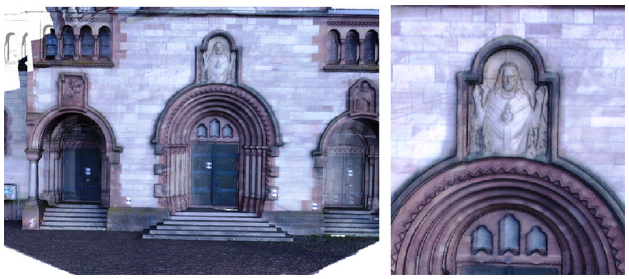


Fig. 8 Texture mapping of Herz-Jesu-K7

Table 3 SfM versus our method (two views)

	Method	Fountain	Herz-Jesu
$\Delta R'$(RMS)	SfM	0.0044	0.0072
	Our method	8.49e−4	0.0013
$\Delta t'$(RMS)	SfM	0.0404	0.0757
	Our method	0.0031	0.0052
3D error	SfM	0.0011	0.0025
	Our method	5.95e−4	0.0018

assumes that images are asynchronized w.r.t. the 3D sensor. In fact, it is equivalent to inaccurate extrinsic of images w.r.t the sensor.

Figure 8 shows that the provided camera poses are very satisfactory. First, the 3D reconstructions for every consecutive pair of images are obtained using classical SfM. All these results are then refined separately using our method. Results before and after the refinement are compared against the ground truth in Table 3. The 3D errors shown here are the mean 3D RMS error of all the pairs. During the implementation, we have decimated the 3D scenes to about 50,000 points by uniform down-sampling for a faster computation. About 2000–3000 feature points were selected in each pair of views for the reconstruction.

For the multiview case, reconstructions from each consecutive pair of views are registered. Such registration undergoes error accumulation and scale factor drift. We separately refined these results using our method and sparse BA (Lourakis and Argyros 2009). The results using our method were found to be significantly better than those of BA. We also considered refining our results using BA.

Table 4 BA versus Our method and unsuccessful refinement of our results using Bundle Adjustment—BA (multiview)

	Method	Fountain	Herz-Jesu
$\Delta R'$(RMS)	BA	0.0436	0.0123
	Our method	0.0020	0.0067
	Refined	0.0251	0.0080
$\Delta t'$(RMS)	BA	0.0311	0.0402
	Our method	0.0019	0.0224
	Refined	0.0172	0.0241
3D error	BA	0.0020	0.0069
	Our method	0.0015	0.0068
	Refined	0.0020	0.0069

Fig. 9 Texture mapping: Bundle Adjustment (left), our method (right)

Results obtained from BA, our method, and BA performed to refine our results are shown in Table 4. It is observed that BA performed on our results diverges from the ground truth instead of further refinement. Since BA takes only the image information into account and cannot incorporate the 3D knowledge, noise present in the image might be the reason for BA to diverge. As the efficiency of BA depends upon the number of observations (images in this case), the difference between BA and our method becomes wider with the decrease in number of observations. This effect can be seen by comparing multiview and two views cases, between Tables 4 and 3.

For qualitative analysis, results obtained from BA as well as our method were used to map the texture (Fig. 9). Texture mapping using BA contains many artifacts the most visible of which has been circled in this figure. Note that, as the scene being relatively far from the cameras, even a small error in pose can significantly affect the texture mapping. It clearly shows the pose refinement using our method is very accurate and visually no different from the ground truth.

5.2.2 Synchronous case

We have also tested our method using two different real and synchronous datasets. Both datasets were acquired by a moving vehicle equipped with a laser-camera system. However, these two setups greatly differ from one another.

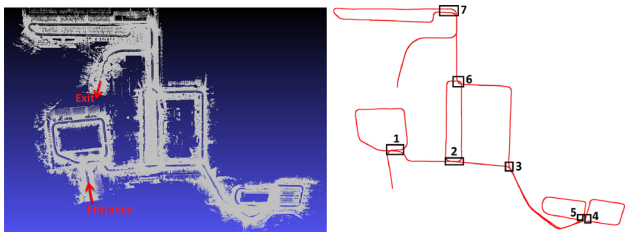


Fig. 10 Large map reconstructed using Laser-Camera system in a single trip shown with starting and end points (left). Closed loops made during the travel. Boxes shown are the loop closing locations of seven different loops (right)

KAIST dataset We conducted our first Synchronous experiment using data obtained from a Laser-Camera system dedicated to reconstructing very large outdoor structures. This system uses two 2D laser scanners and four 2D cameras which are synchronized and calibrated for both intrinsic and extrinsic parameters. Laser scanners used here provide a wide angle of view of the scanning plane so that the system can observe tall objects as well as the ground making it suitable to scan the environment from a close distance. The 3D map (reconstruction) of the environment is made by collecting these 2D scans at their proper location. Therefore, this system requires a very precise localization for a good reconstruction. Extrinsic parameters of 2D cameras were estimated by laser points and a pattern-based calibration method. However, it still possesses a mean projection error of about 0.5 pixels. The interested reader may refer to Bok et al. (2011) for details regarding the experimental setup. The dataset we have tested is a continuous trip of the Laser-Camera scanning system within the compound of KAIST (Korea) for a distance of about 3 KM. The system made seven different loops during its travel. The original reconstruction and the loops are shown in Fig. 10. The lengths of the loops, as shown in Table 5, range from about 200 m to 1.5 KM. Each camera captured 480 × 640 pix. images with a rate of about 20 frames/s. The 2D-to-2D correspondences are computed between images escaping each 10 frames. The original reconstruction obtained by the Laser-Camera system was used as the required 3D information for our method. Note that this reconstruction was not very accurate. Nevertheless, we were still able to refine the motion using such inaccurate data.

The qualitative and quantitative results are presented in Fig. 11 and Table 5 respectively. The results are compared against (Bok et al. 2011) that uses 2D–2D-based refinement method. The errors were computed by performing the ICP between two point clouds captured at the loop closing point before and after the loop travel. Note that loop closing methods are not applied to the presented results. Our goal is to obtain a better localization so that it would be suitable for the loop closing methods. We strongly believe that the localization with such accuracy can be a very suitable

Table 5 Loop size and loop closing errors in meters for Bok et al. (2011) and our method

Loop	Size (m)	Bok et al. (m)	Our method (m)
1	351.76	4.063	1.548
2	386.38	4.538	1.469
3	224.37	4.765	4.398
4	242.87	1.696	1.077
5	931.14	3.884	2.858
6	1496.4	7.182	6.381
7	546.05	5.502	2.115

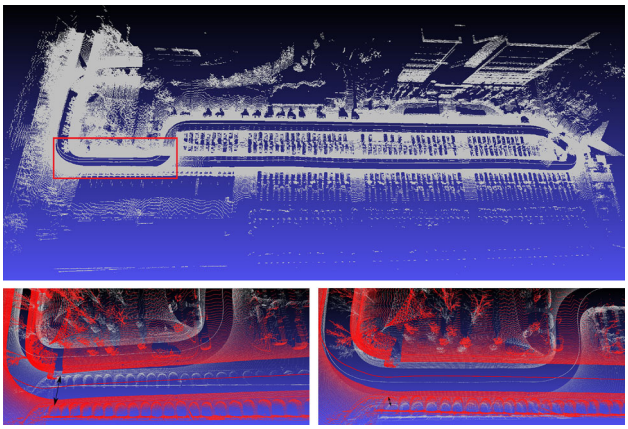


Fig. 11 Results similar to Fig. 1 for seventh Loop. Reconstruction with a red box at the loop closing location (top), obtained using Bok et al. (bottom-left) and our method after refinement (bottom-right). The double sided arrows show the gap between two different reconstructions of the same scene

input for loop closing. The experiments clearly show significant improvements in loop closing errors by our method for all the loops tested. Since, most of the loop closing methods used in practice provide only the local optimal solution; these improvements can contribute to their convergence to the desired one. It can also be seen that the error reduction does not correlate well with the loop length. In fact, the improvement is dependent upon the quality of feature points. The remaining residual error is the combined effect of the errors in calibration, matching, and measurements.

To analyze reconstruction accuracy, we fitted the surface on the reconstructed points cloud using an algorithm that we have developed in-house. This algorithm takes advantage of the camera motion and the order of scanned points. The reconstructed surface was mapped with texture from the same images that were used for localization. The textured scene with its various stages is shown in Fig. 12 for only one side of the reconstruction around the first loop (about 350 m). This part of the reconstruction consists of about 1.3×10^6 3D points and 2.5×10^6 triangles.

KITTI dataset The proposed method was also tested on the benchmark dataset available at (http://www.cvlibs.net/datasets/kitti/). The details of the experimental setup is described in Geiger et al. (2013). We have used the stereo pair of gray images and the 3D data scanned from a Velodyne laser scanner. The results obtained before and after refinement for 5 different sequences were compared against the provided ground truth. Errors in rotation and translation were computed by using the evaluation code provided along with the dataset which uses the ground truth obtained using GPS and other odometry sensors. Although this ground truth might not be very accurate for local poses comparison, it is relevant over a long sequence due to no error accumulation process. Therefore, the errors were measured at the sequence steps of (100, 200, …, 800) and are presented in Table 6. Figure 13 shows the map obtained for the fifth sequence. A close observation shows that the localization before the refinement is already quite satisfactory. Its further refinement makes the result very close to the ground truth itself. Here again, the results are presented without the loop closing.

5.2.3 Cascaded asynchronous-to-synchronous model

We also processed the results obtained from the "Camera alignment" step of asynchronous method using the synchronous data processing algorithm. First, we obtained the poses of first set of asynchronous cameras using (13). Starting from the obtained poses, we used Algorithm 2 to refine the relative poses T_i'. In addition to the EPFL dataset, two sequences from DTU dataset were used for the experiments.

DTU dataset: An industrial robot mounted with two cameras and a projector acquires the scene points using a structured light system. This dataset consists of these scene points and several images along with their precise ground truth poses. Although the detailed information about DTU dataset can be found in Jensen et al. (2014), sample images of the tested sequences are shown in Fig. 14.

Results obtained in each step for EPFL and DTU datasets are shown in Table 7. Figure 15 shows all cameras in the scene for one of the sequences from EPFL dataset. It can be observed that the camera poses obtained from cascaded asynchronous-to-synchronous model are satisfactory. However, they are not always as good as the ones obtained from asynchronous algorithm. This happens mainly because the synchronous algorithm is relatively more sensitive to the pose gaps. In few cases, when the asynchronous algorithm does not produce results very close to ground truth, the synchronous algorithm rather deteriorates the results instead of further improvement. Nevertheless, the absolute poses obtained from the asynchronous algorithm remains unaffected.

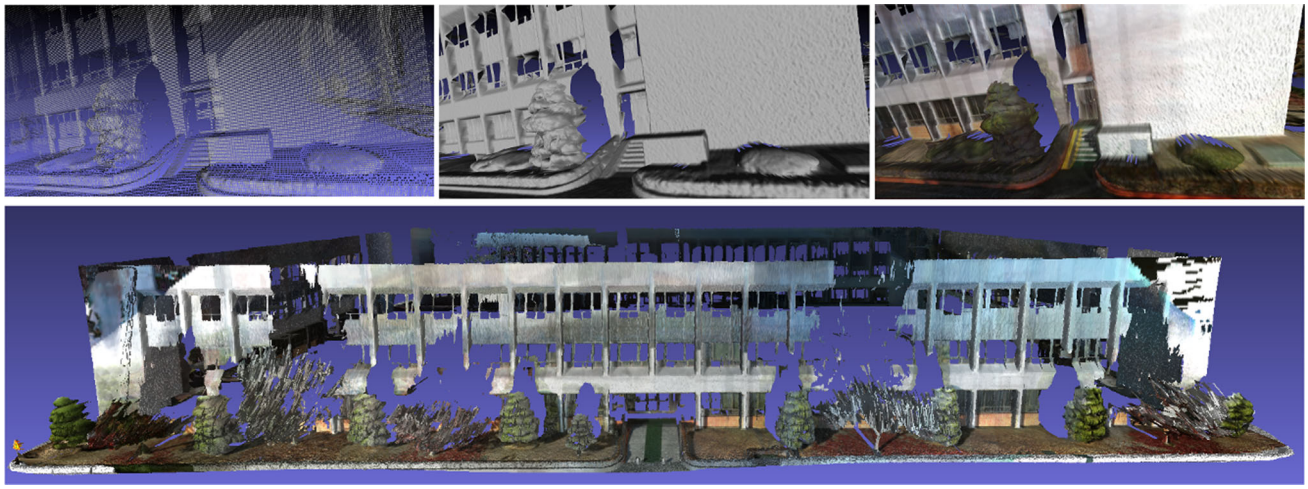


Fig. 12 Surface reconstruction and texture mapping showing the accuracy of localization. Reconstructed 3D, fitted surface, and texture mapping in a close view (top row, left to right). Texture mapping of the structure scanned around loop 1 (bottom)

Table 6 Translation (Δt) and Rotation (ΔR) errors in Initial and Refined results for five different sequences

Sq.N	N.Frames	Initial estimate		Refined	
		Δt (%)	$\Delta R(^\circ/m)$	δt (%)	$\Delta R(^\circ/m)$
3	801	1.6774	0.000432	1.6398	0.000216
5	2761	1.9147	0.000245	1.8679	0.000162
7	1101	2.3410	0.000231	1.5689	0.000192
8	4071	2.3122	0.000447	1.9799	0.000196
9	1591	1.7562	0.000270	1.5604	0.000197

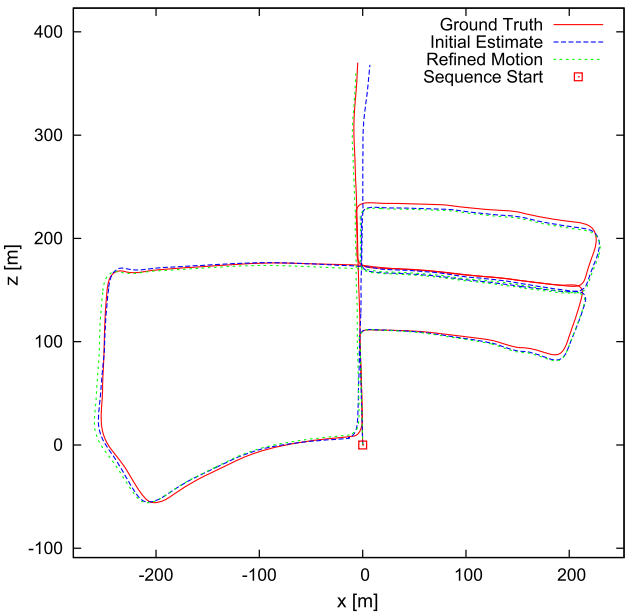


Fig. 13 Map built by our method (Initial Estimate and Refined Motion) vs. Ground Truth for the fifth sequence

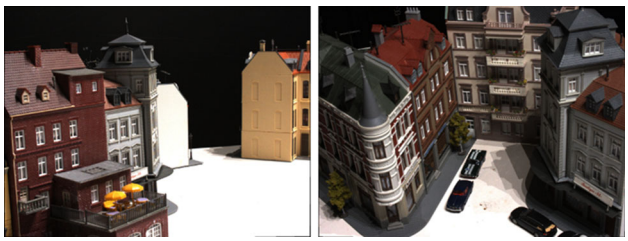


Fig. 14 Sample images from DTU dataset. Left: scan27; right scan73

Table 7 Error measured for cascaded asynchronous-to-synchronous model

	Asynchronous		Synchronous	
	ΔR (mean)	Δt (mean)	ΔR (mean)	Δt (mean)
Fountain-P11	0.0214	0.0074	0.0230	0.0111
Herz-Jesu	0.0222	0.0182	0.0196	0.0191
Scene27	0.0747	0.0373	0.1723	0.0850
Scan73	0.0496	0.0249	0.0479	0.0214

6 Conclusion

A framework to fuse the information from synchronous or asynchronous 2D and 3D cameras for visual odometry has been proposed. Our demonstration with several experiments show the possibility of estimating accurate motion of 2D–3D camera system, even when 2D and 3D cameras are not synchronized and the 3D scene includes some inaccuracies. Usage of 3D scene points to refine the 2D camera poses is the key to achieve such accuracy. To make it possible for asynchronous cameras, a direct 2D-to-3D registration method has also been integrated in the optimization process. The adaptation of proposed framework for synchronous

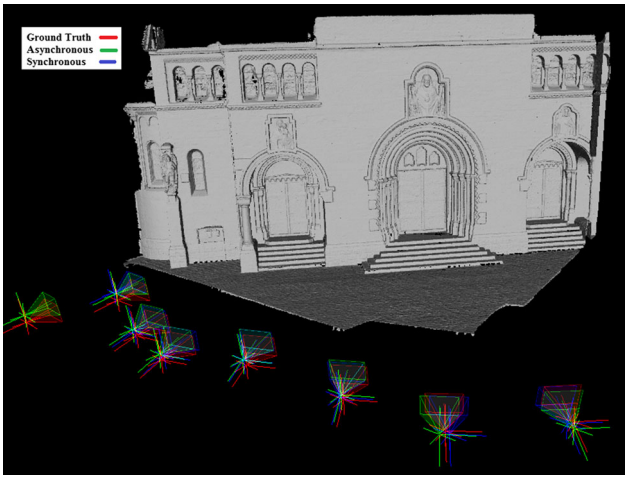


Fig. 15 Ground truth, Asynchronous-to-Synchronous Cameras poses in the scene

cameras, although being straightforward, was found to be very effective for pose refinement. In general, the treatment of asynchronous cameras as asynchronous throughout the process is a better choice over the cascaded asynchronous-to-synchronous model assumption.

References

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis*, *14*, 239–256.

Bok, Y., Jeong, Y., Choi, D. G., & Kweon, I. S. (2011). Capturing village-level heritages with a hand-held camera-laser fusion sensor. *International Journal of Computer Vision*, *94*, 36–53.

Buczko, & Willert, V. (2016). Flow-decoupled normalized reprojection error for visual odometry. In *IEEE Intelligent Transportation Systems Conference (ITSC)*.

Chiuso, A., Favaro, P., Jin, H., Soatto, S. (2000). 3-D motion and structure from 2-D motion causally integrated over time: Implementation, ECCV.

Christy, S., & Horaud, R. (1999). Iterative pose computation from line correspondences. *Computer Vision and Image Understanding*, *73*, 137–144.

Clarkson, M. J., Rueckert, D., Hill, D. L. G., & Hawkes, D. J. (2001). Using photo-consistency to register 2D optical images of the human face to a 3D surface model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 1266–1280.

Comport, A., Malis, E., & Rives, P. (2007). Accurate quadri-focal tracking for robust 3D visual odometry, ICRA.

Corsini, M., Dellepiane, M., Ganovelli, F., Gherardi, R., Fusiello, A., & Scopigno, R. (2013). Fully automatic registration of image sets on approximate geometry. *International Journal of Computer Cision*, *102*, 91–111.

Eckart, B., Kim, K., Troccoli, A., Kelly, A., & Kautz, J. (2015). Mlmd: Max-imum likelihood mixture decoupling for fast and accurate point cloud registration. In *3DVision (3DV), 2015 International Conference on* (pp. 241–249).

Evangelidis, G. D., Kounades-Bastian, D., Horaud, R., & ZPsarakis, E. (2014). A generative model for the joint registration of multiple point sets. In *ECCV* (pp. 109–122).

Fitzgibbon, A. (2003). Robust registration of 2D and 3D point sets. *Image and Vision Computing*, *21*, 1145–1153.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, *32*, 1231–1237.

Henry, P., Krainin, M., Herbst, E., Ren, X., & Fox, D. (2012). RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments, IJRR.

Hesch, J. A., & Roumeliotis, S. I. (2011) A direct least-squares (DLS) method for PnP, ICCV.

Holz, D., Lorken, C., & Surmann, H. (2008). Continuous 3D sensing for navigation and SLAM in cluttered and dynamic environments, ICIF.

Horaud, R., Forbes, F., Yguel, M., Dewaele, G., & Zhang, J. (2011). Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(3), 587–602.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., & Aanaes, H. (2014). Large scale multi-view stereopsis evaluation, CVPR.

Jia, Li, B., Zhang, G., & Li, X. (2016). Improved kinect fusion based on graph-based optimization and large loop model. In *2016 IEEE international conference on information and automation (ICIA)* (pp. 813–818). Ningbo.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., & Dellaert, F. (2011). iSAM2: Incremental smoothing and mapping using the Bayes tree. IJRR.

Kerl, C., Sturm, J., & Cremers, D. (2013). Dense visual SLAM for RGB-D Cameras, IROS.

Knopp, J., Sivic, J., & Pajdla, T. (2010). Avoiding confusing features in place recognition, ECCV.

Koch, R. (1993). Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*, 556–568.

Lhuillier, M. (2012). Incremental fusion of structure-from-motion and GPS using constrained bundle adjustments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 2489–2495.

Liu, L., & Stamos, I. (2005). Automatic 3D to 2D registration for the photorealistic rendering of urban scenes, CVPR.

Lourakis, M. I. A., & Argyros, A. A. (2009). SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, *36*, 2.

Martin, K., & Jakob, W. (2012). *Iterative closest point*. Lyngby: Technical University of Denmark.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohli, P., Shotton, J., Hodges, S., & Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking, ISMAR.

Nister, D. (2004). A minimal solution to the generalised 3-point pose problem, CVPR.

Nister, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 756–770.

Nüchter, A., Lingemann, K., Hertzberg, J., & Surmann, H. (2007). 6D SLAM-3D mapping outdoor environments: Research articles. *Journal of Field Robotics*, *24*, 699–722.

Nister, D., Naroditsky, O., & Bergen, J. (2004). Visual odometry, CVPR.

Paudel, D. P., Demonceaux, C., Habed, A., & Vasseur, P. (2014). Localization of 2D cameras in a known environment using direct 2D-3D registration, ICPR.

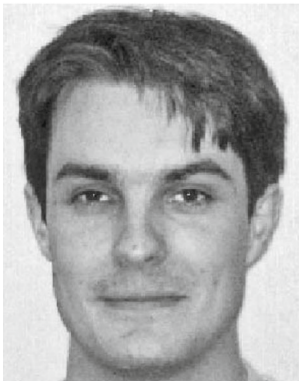
Paudel, D. P., Demonceaux, C., Habed, A., Vasseur, P., & Kweon, I. S. (2014). 2D-3D camera fusion for visual odometry in outdoor environments, IROS.

Pire, T., Fischer, T., Civera. J., De Cristóforis, P., & Berlles, J. J. (2015). Stereo parallel tracking and mapping for robot localization. IROS.

Pomerleau, F., Colas, F., & Siegwart, R. (2015). A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, *4*(1), 1–104.

Ramalingam, S., Bouaziz, S., Sturm, P. & Brand, M. (2009). Geolocalization using skylines from omni-images, ICCV Workshops.
Rawia, M., Vasseur, P., Mousset, S., Boutteau, R., & Bensrhair, A. (2014) Visual odometry with unsynchronized multi-cameras setup for intelligent vehicle application. In *Intelligent vehicles symposium proceedings*.
Rusinkiewicz, S., & Levoy, M. (2001) Efficient variants of the ICP algo-rithm, 3DIM.
Sattler, T., Leibe, B., & Kobbelt, L. (2011). Fast image-based localization using direct 2D-to-3D matching, ICCV.
Smith, A. E., & Coit, D. W. (1995). *Penalty functions*. Pittsburgh: University of Pittsburgh.
Stoyanov, T., Magnusson, M., & Lilienthal, A. J. (2012). Point set registration through minimization of the L2 distance between 3D-NDT models. In *IEEE International Conference on Robotics and Automation* (pp. 5196–5201).
Taguchi, Y., Jian, Y. D., Ramalingam, S., & Feng, C. (2013). Point-plane SLAM for hand-held 3D sensors. ICRA.
Tamaazousti, M., Gay-Bellile, V., Collette, S. N., Bourgeois, S., & Dhome, M. (2011). NonLinear refinement of structure from motion reconstruction by taking advantage of a partial knowledge of the environment, CVPR.
Taneja, A., Ballan, L., & Pollefeys, M. (2012). 3DIMPVT, registration of spherical panoramic images with cadastral 3D models.
Tardif, George, M., Laverne, M., Kelly, A., & Stentz, A. (2010). A new approach to vision-aided inertial navigation. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 18–22).
Trevor, A. J. B., Rogers, J. G., & Christensen, H. I. (2012). Planar surface SLAM with 3D and 2D sensors, ICRA.
Triggs, B., Mclauchlan, P., Hartley, R., & Fitzgibbon, A. (2000). Bundle adjustment a modern synthesis. Vision Algorithms: Theory and Practice, LNCS.
Viola, P., & Wells III, W. M. (1997). Alignment by maximization of mutual information. *International Journal of Computer Vision*, *24*, 137–154.
Weingarten, J. W., Gruener, G., Siegwart, R. (2004). A state-of-the-art 3D sensor for robot navigation, IROS.
Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., & Tardós, J. (2009). A comparison of loop closing techniques in monocular SLAM. *Robotics and Autonomous Systems*, *57*, 1188–1197.
Zhang, J., Kaess, M., & Singh, S. (2014). Real-time depth enhanced monocular odometry. Intelligent Robots and Systems, IROS.
Zhao, W., David, N., & Steve, H. (2005). Alignment of continuous video onto 3D point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1305–1318.

Danda Pani Paudel is a postdoctoral researcher in the Computer Vision Lab at ETH Zurich. His research interests include visual-SLAM, mathematical optimization, and geometric problems in computer vision. Currently, he is working in the field of vision-based city scale traffic flow modeling. Danda Pani received a Doctor of Philosophy (Ph.D.) in 2015, and a Master's degree in computer vision in 2012, from University of Bourgogne, France. He also worked as a research scholar at University of Strasbourg, France, from 2013 to 2015, while devising global methods for 2D-3D registration problems.



Cédric Demonceaux received the M.Sc. degree in Mathematics in 2001 and the Ph.D. degree in Image Processing from the Université de Picardie Jules Verne (UPJV), France, in 2004. In 2005, he became associate professor at MIS-UPJV. From 2010 to 2014, he obtained a CNRS-Higher Education chair at Le2I UMR CNRS, Université de Bourgogne. Since 2014, he is full Professor at the University of Burgundy. His research interests are in image processing and computer vision for robotics applications.

Adlane Habed is an Associate Professor of Computer Science at the University of Strasbourg (France) and a member of the ICube (CNRS) research laboratory. He received a Ph.D. in Computer Science from the University of Sherbrooke (Canada) in 2005. From 2001 to 2007, he served as a Computer Science Full-time Lecturer (2001 - 2005) and as an Assistant Professor (2005 - 2007) at the University of Windsor's School of Computer Science (Canada). Prior to his current appointment, he held an Assistant Professor position, from 2007 to 2012, at the University of Bourgogne (France) and was member of the Le2i (CNRS) research laboratory. His research interests are in the field Computer Vision and Optimization.

Pascal Vasseur received the MS degree in System Control from the Université de Technologie de Compiégne, France, in 1995, and the Ph.D. degree in Automatic Control from the Universite de Picardie Jules Verne, France, in 1998. He was an Associate Professor at the University Institute of Technology of Amiens between 1999 and 2010. He is now a professor at the University of Rouen in the LITIS Laboratory. His research interests include Computer Vision and its applications to mobile and aerial robots.