Vision, Modeling, and Visualization (2016)
M. Hullin, M. Stamminger, and T. Weinkauf (Eds.)

Fast Metric Acquisition with Mobile Devices

Valeria Garro[2], Giovanni Pintore[1], Fabio Ganovelli[2], Enrico Gobbetti[1], Roberto Scopigno[2]

[1]Visual Computing Group, CRS4, Italy†
[2]Visual Computing Group, ISTI CNR, Italy‡

Abstract

We present a novel algorithm for fast metric reconstruction on mobile devices using a combination of image and inertial acceleration data. In contrast to previous approaches to this problem, our algorithm does not require a long acquisition time or intensive data processing and can be implemented entirely on common IMU-enabled tablet and smartphones. The method recovers real world units by comparing the acceleration values from the inertial sensors with the ones inferred from images. In order to cope with IMU signal noise, we propose a novel RANSAC-like strategy which helps to remove the outliers. We demonstrate the effectiveness and the accuracy of our method through an integrated mobile system returning point clouds in metric scale.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Computer Graphics]: Image processing and computer vision—Scene analysis

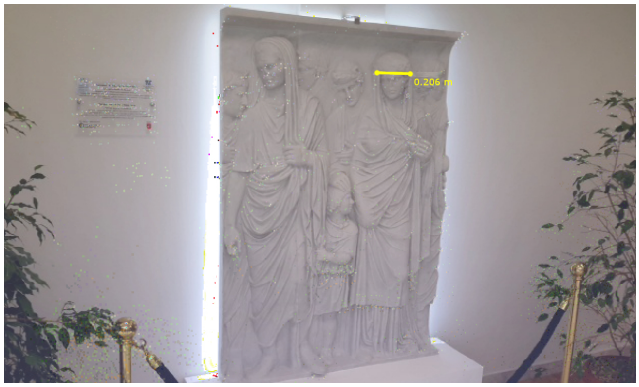

Figure 1: *Metric measurement of an* Ara Pacis Flamini's *frieze (marble copy for public display) acquired by our system. The measurement has been taken by the user on the reconstructed 3D point cloud and returned to display in metric dimension. The average error compared to ground truth is 2.8%.*

1. Introduction

3D shape digitization is a hot research topic with many applications. Depending on the size of the object of interest, the required accuracy, the time and the money to invest, one can choose among a quite wide range of software and hardware solutions. Among these, in the last ten years, image-based acquisition techniques such as Structure from Motion (SfM) have become a popular tool. This is most likely due to two factors: the increasing computational power of common devices, that made these demanding algorithms practical, and the fact that they do not require specialized acquisition hardware, since photographs are nowadays easy to get with off-the-shelf smartphones or from the internet. However, one weakness of image-based methods is that their output is intrinsically up-to-scale, that is, they have no real world dimension.

The most straightforward way to overcome scale ambiguity is to provide at least two points in the 3D reconstruction that are separated by a known distance, so that the scale factor can be recovered. This can be easily achieved by using markers or measuring real-world distances between at least two physical points in space that are also in the reconstruction. This approach, however, is data dependent and, as such, not applicable to the general case.

With the introduction of onboard inertial measurement unit (IMU) sensors and high precision clocks on mobile devices, photos taken with a smartphone or a tablet device can be accompanied with information on how the device moves from pose to pose. This should make in principle feasible to compute a scale factor between image-based reconstructions and real world units. Although in theory one could use double integration of the acceleration vector to compute such a trajectory in real world units, most IMU sensors, and especially those mounted on commonly available devices, provide a very noisy signal, especially for small accelerations, and cannot be reliably used with this approach. The problem is well known, and it generated a consisting body of literature, especially

† http://www.crs4.it/vic/
‡ http://vcg.isti.cnr.it

DOI: 10.2312/vmv.20161339

in the field of robotics, for example for unmanned vehicle navigation, reconstruction from aerial images, or SLAM.

All the proposed solutions in the 3D capture domain (see Sec. 2) cope with this problem by assuming that the acquisition lasts a relatively long period of time (typically several minutes) and consists of hundreds or thousands of images. This approach is applicable when the goal is to obtain an accurate and dense reconstruction, but becomes too costly and time consuming when the goal is just to quickly recover the structure and shape of objects (for instance, for acquiring metric furniture shapes for indoor 3D plan generations).

Approach. We propose a system that only requires making a quick video sequence of the object of interest captured by a moving camera and it returns a 3D point cloud in real world units. The basic idea beside our method is, instead of deriving positions from acceleration values from the IMU, to register IMU acceleration values with the accelerations of the camera inferred from images. In our approach, summarized in Sec. 3, 3D reconstruction is carried out by an incremental SfM implementation. We track image features over all the acquired frames and use them for triangulation when the estimated baseline formed by the corresponding camera position is large enough. Then, we use the reconstructed 3D points to solve the Perspective-n-Point (PnP) problem for every frame, thus obtaining a dense sampling of all camera poses. The camera trajectory is finally defined by using those camera poses as Catmull-Rom spline's control points, from which acceleration values can be computed at any point. Alignment with the IMU accelerations is achieved with a robust RANSAC-based algorithm, which helps to remove the outliers in the IMU values, which in this context mostly means sudden peaks of acceleration. The details of reconstruction algorithm and how vision-based camera trajectory is computed are given in Sec. 4, while Sec. 5 describes the process of recovering the scale factor. An evaluation of the method in a real-world setting is presented in Sec. 6.

Contribution. We formulate the metric scaling problem as a derivation task, rather than an integration problem, avoiding the considerable error encountered by online schemes (e.g., [TKM*13]) when integrating noisy, biased accelerometer measurements. In addition, our approach does not need a device orientation measurement (i.e. a compass), which is even more prone to error than accelerometer. Moreover, by exploiting our specialized SfM pipeline and interpolation track to match the IMU samples with each video frame pose, we maximize the number of samples usable for scale estimation, contrary to matching approaches based on the IMU samples downsampling (e.g., [HLS14]). In order to robustly perform the mapping, we introduce a novel RANSAC-based approach which robustly finds a scaled rigid body map in acceleration space, resulting in a more accurate samples filtering and robust scale estimation, in contrast to minimization approaches based only on the scale factor (e.g. [HLS14, JT01]).

Advantages. To the best of our knowledge, this is the first method for providing a metric reconstruction from few seconds of acquisition time, without requiring initialization or landmarks. Given the light-weight approach, we enable users to perform fast metric acquisition of 3D shapes exploiting the capabilities of modern mobile devices such as processing and sensors fusion.

2. Related Work

The combination of inertial and visual sensors has been used for a long time, well before the inception of modern smartphones. A consistent body of literature, mostly to be found in the field of robotics, shows widespread effort to overcome the intrinsic ambiguities of monocular acquisition by adding inertial data in applications like SLAM. The many studies on the subject consider different hardware settings (e.g., cameras mounted in cars, planes or robots) and different goals (3D reconstruction, localization, unmanned navigation), and a complete overview is well beyond the scope of this paper. In the following, we will concisely summarize the state of the art and more extensively cover the most recent contribution that are more closely related to our approach.

In the most general terms, the problem is posed as finding an estimation of all the variables of a system described by camera and IMU. These variables typically include position, velocity, acceleration and biases but of course, the exact formulation varies with the sensors provided. The *observability* of a variable indicates how well that variable can be inferred by the external outputs. For example, visual input or inertial input alone are not sufficient for making the motion (and hence the scale factor) observable, and, as shown by Jones et al. [JS11], even with visual and inertial information combined the condition of non-zero linear acceleration must hold. An in-depth study of observable quantities in vision and IMU system is provided by Martinelli et al. [Mar12], where closed-form solutions for observable quantities from the data output collected in a very short time interval are also provided.

Several approaches apply filtering methods, that is, they refine the estimation of variables over a large number of states by applying the ubiquitous (Extended) Kalman Filter [Kal60]. Examples include improving vision-only SLAM [PLST07, LS08], supporting autonomous navigation [HR01, TGL*10], or motion estimation in virtual reality applications [Cha01, PRCZ12]. Filtering methods only need to store the last state and the current state of the system, so they can be used online (which is a mandatory condition for application domains such as virtual reality) if the set of features (which are also part of the state) is small. Conversely, the new features are added in the process, long-term runs become unfeasible.

Other approaches gather all the data and use offline optimization techniques. GPS and image data are often combined to obtain a more accurate structure from motion reconstruction by minimizing the re-projection errors of 3D points using Bundle Adjustment [Lhu12, SFF14, JEJR04, FPL*10]. A more recent work [APS15] improves the SfM implementation with a pipeline tailored to the case where images are assumed ordered and GPS and IMU sensors are available, named BA4S (Bundle Adjustment for Sequential Imagery). The precision of batch optimization and efficiency of filtering are often combined by keeping track of a set of most recent states, that is, not only the last one like in filtering and not all of them like in BA [DSM11, LLB*14, FCDS15]. The approach presented in [WS11] on the Extended Kalman Filter SLAM, avoids including features in the state vector by treating the vision framework as a black box from which it takes poses estimation (and covariance matrix for prediction) obtaining a constant update time. Drift and pose estimation error are handled by detect-

ing abrupt changes on their value (while pose is assumed to change smoothly).

The approaches cited so far are not directly aimed to compute the scale factor but, instead, strive to improve the quality of the results (of SLAM, odometry, navigation etc.) harnessing inertial sensors. In Tanskanen et al. [TKM*13], an online implementation of a 3D metric acquisition pipeline is presented, estimating the device acceleration by integration using Velocity Verlet. This method is a filtering approach based on Kalman filter and the scale is computed by using an event-based approach that records swift movements to estimate the scale factor with larger acceleration values. As best result on the scale factor estimation it obtains an error between 10% to 15% with a reference test object (a cylinder).

Our core idea is more related to methods that have been introduce to recover metric camera motion [JT01, HLS14]. In such approaches, SfM is used to estimate camera orientation and position (up to scale). Thanks to temporal marks on the images, the acceleration of the camera can be derived correspondingly to the image time. So the problem is posed as minimizing the differences between the acceleration derived from SfM for the temporally marked images and the acceleration obtained by the inertial sensor. In Jung et al. [JT01] such difference is minimized with respect to that parameters of a series of splines, and similarly in Ham et al. [HLS14] minimizing directly the scale and a bias factor, after the IMU signal is downsampled and antialiased with a convolution matrix. Due to the sparse correspondences between camera and downsampled IMU acceleration, the authors also add an alignment phase to register IMU and vision signal exploiting their cross-correlation. By contrast, in our approach we maximize the number of correspondences integrating the SfM pipeline in the scale estimation process and exploiting the 2D features matching to recover all the camera poses related to IMU samples. In this way we enable the introduction of a more accurate samples filtering and scale estimation by robustly fitting a similarity transform and not only a single scale factor.

3. Approach overview

We introduce a specialized mobile reconstruction pipeline (Fig. 2) to capture the SfM scene coupled with the inertial measures, summarized in two steps. In the first step, the user takes a temporally indexed video of the object by moving the device so to obtain a sufficiently large baseline for the stereo processing, that typically is a horizontal translation in front of the object. Simultaneously the values of the device's linear acceleration provided by the IMU are stored with their corresponding acquisition timestamps. From the obtained set of frames registered on the same multi-view scene reference system, we estimate the pose of each corresponding camera by using the 2D-3D correspondences between tracked 2D features and computed 3D points. Each camera position becomes then a control point of an *interpolation track* of rigid body maps.

In the second step, for each IMU sample acquired we calculate the corresponding camera acceleration by finite differences, we compare then the body-centric linear acceleration of the device with the estimated camera acceleration.

Once a matching between device and camera body-centric accelerations is established we introduce a novel RANSAC approach to obtain a direct and robust estimation on the metric scale, looking for a scaled rigid body map between the two acceleration sets in the acceleration space (Fig. 3). This strict alignment hypothesis combined with a MLESAC robust estimator strategy provides greater robustness to error, as shown in the results Section 6.

4. On-the-fly point cloud reconstruction

The reconstruction pipeline runs in real time on device and it is composed by two main modules which run simultaneously during the acquisition process: the `inertial module` and the `vision module`.

The inertial module reads and stores the values of the device's linear acceleration provided by the IMU sensor and the corresponding acquisition timestamps. The coordinates system of the linear acceleration vector is defined relative to the screen of the phone.

The second module is the vision module, its main goal is to track local features extracted from the first frames of the entire video to obtain the camera pose estimation of the video frames and the sparse points cloud reconstruction of the target object that the user wants to acquire. The vision module can be divided into two stages: the first stage, in which each video frame is captured and processed in order to have a partial 3D points cloud of the object in a dedicated reference system; the second stage consisting in the alignment and merging of all the 3D points clouds obtained in the first stage and the global camera poses estimation of all the video frames in a unique reference system.

During the first stage the algorithm extracts Shi-Tomasi features [ST94] from the first frame and tracks them along the following frames using Lucas-Kanade optical flow method [LK81]. Please note that we choose the features which empirically led to the best results for our mobile setup and images, the proposed method remains of course equally valid also with alternative SfM pipelines. Dealing with video frames we can exploit the continuity of the video sequence and keep track of the features extracted from the first frame f_0 during the entire sequence. For each incoming frame f_i, if a sufficiently large baseline is detected (by measuring the motion of the tracked features) f_0 and f_i are eligible for the Fundamental matrix estimation. The Essential matrix can be computed knowing the intrinsic parameter of the camera and so the rotation and translation matrices RT of the second frame with respect to the fixed camera reference $[I|0]$ of frame f_0 are found.

Once the relative position of the two cameras is found, we perform a triangulation of the corresponding 2D features obtaining a 3D points cloud. These processing steps are computed in real time during the video acquisition for all the N video frames f_i with $i = 1,\ldots,N$. Furthermore, the data related to the tracked features (2D correspondences between features through the N frames and 2D-3D correspondences between features and 3D points) are stored. Note that at the end of the video shoot just a subset of $M \leq N$ frames will be associated with a point cloud considering that some of the aforementioned steps could not end with a positive result (e.g., the baseline could be too small, the Fundamental matrix extraction could fail). After the acquisition, the application automatically starts the alignment of the set of M 3D points clouds.

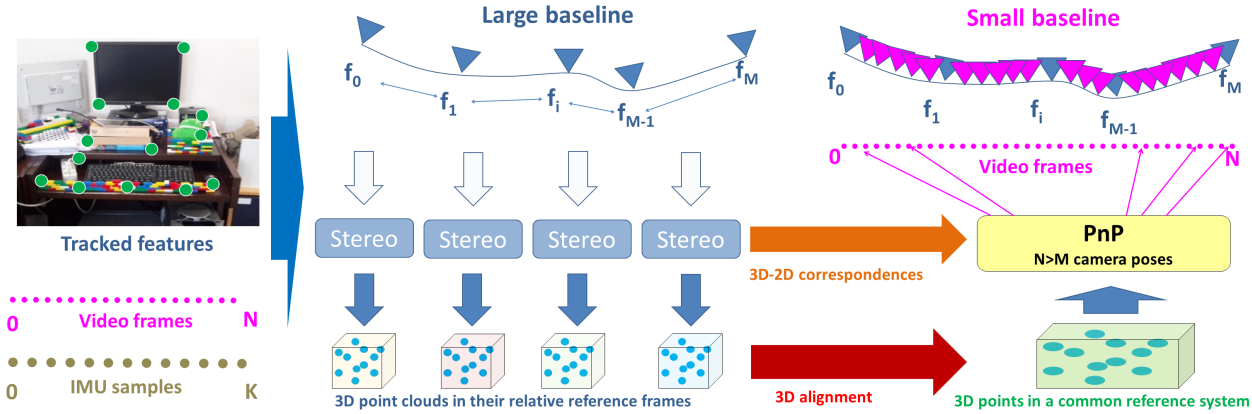

Figure 2: *Scheme of vision module pipeline. Starting from the left. Shi-Tomasi features (illustrative green points) are tracked over the frames; a subset of M frames forms a baseline wide enough to create M point clouds, each in its own reference system. Once the M point clouds are aligned in a single reference system, exploiting the point-to-point correspondences, we run a Perspective-n-Point algorithm on 3D points and their known projection on the frames, resulting in $\tilde{N}$ registered frames (actually $M \leq \tilde{N} \leq N$).*

The algorithm selects a reference cloud by minimizing the geometrical error obtained by aligning to it all the other points clouds using the absolute orientation method of Umeyama [Ume91].

At this point all points clouds have been registered to the same reference system, the final step consists of the camera pose estimation of all the video frames (in the global reference system) in order to maximize the number of registered frames. If among the data related to frame f_i there are a sufficient number of correspondences between its 2D tracked features and a subset of 3D points from the reference cloud we can compute the camera pose of f_i solving the related Perspective-n-Point (PnP) camera pose problem exploiting RANSAC method. The final number $\tilde{N}$ of registered frames will be $M \leq \tilde{N} \leq N$.

5. Recovering the scale factor

To recover the scale factor we adopt a RANSAC strategy, comparing the camera poses estimated at Section 4 with the body-centric inertial measurements of a smart device. Current smartphones are equipped with a 3D gyroscope and accelerometer, which produce (in contrast to larger inertial measurement units) substantial time-dependent and device-specific offsets, as well as significant noise.

We take as input data the linear accelerations relative to the device body A_s temporally indexed by the time instants $\{t_0, \cdots, t_K\}$:

$$A_s = \begin{pmatrix} a_s^x(t_0) & a_s^y(t_0) & a_s^z(t_0) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ a_s^x(t_K) & a_s^y(t_K) & a_s^z(t_K) \end{pmatrix} \quad (1)$$

Contrary to other previous approaches (i.e., [TKM*13]) we do not integrate these data to estimate a spatial device trajectory (this operation is prone to considerable error), neither do we need to use further orientation data from the IMU (i.e. absolute orientation), usually subject to even more errors.

From the vision pipeline we take as input data the temporally indexed positions of each camera, represented as rigid body maps $C_{3x4} = [R_i|t_i]\ i = 1, \ldots, \tilde{N}$. We use these data to estimate the instantaneous linear camera accelerations, and then find the alignment with the IMU accelerations collected in the inertial module (described in Section 4). As similar approaches have proven [HLS14, JT01], replacing the integration operation with the derivative avoids accumulation errors.

Since we expected that the estimation of the scale factor should converge to a correct value with more data, one limit of this approach is the need of a high and uniformly spaced number of calibrated viewpoints. Ham et al. [HLS14] try to get this condition through a very long acquisition time (often many minutes). Nevertheless, many of these samples are lost through an operation of downsampling, necessary in their solution to synchronize the IMU and camera samples (common devices typically record samples at 100 Hz for the IMU and at 30 Hz for the camera). Moreover, this solution assumes that the camera motion is estimated through a time consuming off-line pipeline only after performing a specific calibration of sensors bias and camera-IMU transformation. Jung and Taylor [JT01] follow a similar approach for a robotics setup rather than commodity mobile acquisitions, as it requires high accuracy IMUs and manually calibrated omnidirectional video streams. This led them to assume, contrary to our approach, which is tuned for mobile settings, that the inertial data is more accurate than stereo camera calibration.

To overcome these limitations and achieve good results in the mobile realm, we introduce an *upsampling* strategy to maximize the number of samples from the vision module, as illustrated in Sec. 4, with the goal of supporting a statistical approach to the scaling problem. In this approach, the rotation matrix R that relates the IMU and the camera in the acceleration space (calculated with external tools in previous approaches) is assumed as unknown together with the scale factor. Including R in the computation makes the method more robust, since the error in metric estimation on a

mobile device is strongly affected by outliers and it can not be easily represented by a specific model because of a variety of factors, such as noise varying with device temperature, indoor environment interferences, and random mismatches between sensors and camera stream.

Starting from an initial set of cameras C_M registered on a same scene reference system (multi-view scene coordinates), we use the whole set of N tracked frames ($N > M$) to estimate the intermediate camera positions of $\tilde{N}$ registered views.

Assuming these $C_{\tilde{N}}$ camera positions are temporally indexed on the same temporal reference of the IMU acceleration samples, we exploit them to create *Catmull-Rom* control points of an *interpolation track* of rigid body maps $I_C(t)$, with t indicating time.

For each sample $\{a_s(t_0), \cdots, a_s(t_K)\}$ we search for a temporal match $I_C(t_k)$ in the interpolation track. Considering the position of $I_C(t_k)$ in scene coordinates $p_c(t_k)$ we calculate the acceleration at the instant t_k through 8-th order central finite differences:

$$p_c{}''(t_k) = \frac{\sum_{i=0}^{8} (-1)^{(i+1)} \delta_i * p_c(t_{k+i-4})}{\Delta t^2} \tag{2}$$

with $\Delta t = 1ms$ and the δ_i coefficients chosen according to [For88].

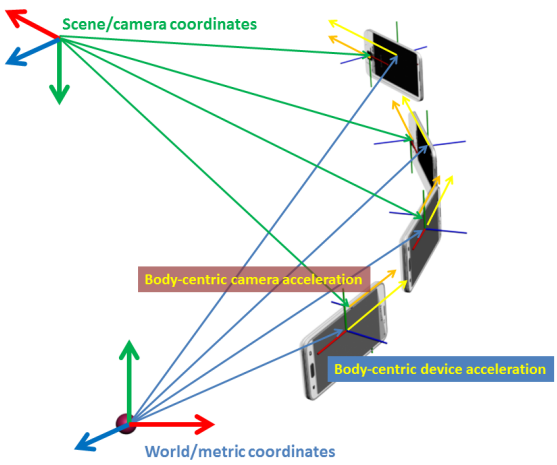

Figure 3: *The SfM algorithm returns the absolute position of the camera in scene coordinates (green position vectors) whereas the absolute position of the device in world coordinates is unknown (light blue position vectors). Since the IMU accelerations (yellow) are in local, body-centric coordinates (relative to the IMU micro-circuit), we need to rotate (through Eq. 3) each camera acceleration to obtain a body-centric acceleration of the camera (orange).*

Since the SfM algorithm returns the position and orientation of the camera in scene coordinates and the IMU measurements are in local body-centric coordinates, to compare them we need to orient the accelerations estimated for the frames to the acceleration values

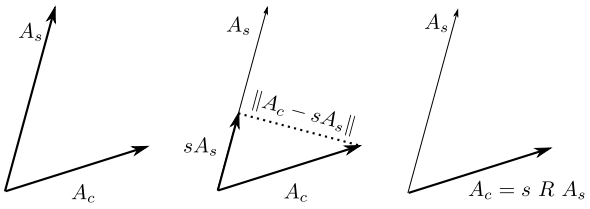

Figure 4: *Introducing rotation R accounts for constant bias and allows any choice for the base of RANSAC algorithm to provide a solution.*

provided by the IMU. We rotate then each $p_c{}''(t_k)$ to obtain the body-centric acceleration of the camera $A_c(t)$:

$$A_c = \begin{pmatrix} p_c{}''(t_0)^T R_c(t_0) \\ \cdot \\ \cdot \\ p_c{}''(t_K)^T R_c(t_K) \end{pmatrix} \tag{3}$$

where $R_c(t_k)$ is is the orientation of the camera in scene coordinates at the t_k instant.

Once a matching between the K device acceleration samples A_s and the K camera acceleration samples A_c is established we introduce a specific RANSAC approach which robustly finds a similarity transform such that

$$\underset{s\ R}{argmin} \{\|A_c - sRA_s\|\} \tag{4}$$

where s is the scale factor between scene coordinates and real world and R is the rotation between the two coordinate systems in the acceleration space (Fig. 3). Please note that A_c and A_s are both expressed in device body coordinates and so in principle R should be the identity transformation. R accounts for constant bias in IMU vector orientation and provides the degrees of freedom to define a similarity for any pair of corresponding vectors. Referring to Figure 4: acceleration vectors A_c and A_s are not collinear and, as such, scaling alone is not enough to transform one into the other. This is different from approaches like Ham et al. [HLS14], where the acceleration vectors are compared imposing only the scale factor, and provides greater robustness to IMU error (see results in Section 6).

To estimate the error we follow the Lo-RANSAC strategy described by Capel et al. [Cap05], adopting as size for the basis set $B_s = 3$. We apply then a MLESAC robust estimator [TZ00], to maximize the likelihood rather than just the number of inliers.

In our specific case, given the true fraction of in-lying correspondences ε, the probability of selecting a basis set of size B_s that consists entirely of inliers is ε^{B_s}. Hence the probability of sampling K basis sets all of which are polluted by at least one outlier is given by

$$\eta = (1 - \varepsilon^{B_s}) \tag{5}$$

Choosing an appropriate confidence threshold $C_{th} = 0.01$ (proven

to be valid for all tested cases), the estimate iterations n_{max} from number of inliers are:

$$n_{max} = \frac{\ln(\eta)}{ln(\varepsilon)+0.5} \tag{6}$$

This bail-out strategy makes the computation very light-weight, making the method suitable for mobile computing. The whole strategy results effective and accurate in many real-world cases, presented in detail in Section 6.

6. Results

We implemented an Android application (compatible with version 4.4 and higher) for the sensor and video acquisition and the 3D reconstruction, testing it on different commodity devices, such as: HTC One M8 with Quad-core 2.3 GHz 2GB RAM, Samsung Galaxy TAB4 with Quad-core 1.2 GHz 1.5GB RAM, Samsung Galaxy Note 10.1 with 1.9GHz Quad-core + 1.3 GHz Quad-core and 3GB RAM. The application has been written in Java and C++ using Android SDK, NDK, and OpenCv4Android [Ope15] as supporting libraries. As a proof-of-concept, the system returns a simple 3D point cloud coupled with the relative camera poses aligned to the same reference system and a scale ratio value between real-world metric space and scene coordinates. The processing time both for the SfM reconstruction and for the metric scale estimation has been negligible.

Tab. 1 summarizes the results obtained for objects whose real dimensions are known. We consider as ground truth the scale ratio between *meters* and *scene units*, manually measured on the point cloud returned by the SfM pipeline [CCR08]. We sampled the data from IMU at different rates between $8ms$ to $200ms$ (four rates are available on the Android API: SENSOR_DELAY_FASTEST: $8-20ms$, SENSOR_DELAY_GAME: $35-40ms$, SENSOR_DELAY_UI: $85-90ms$, SENSOR_DELAY_NORMAL: $215-230ms$).

In the last column of Table 1, we report the results obtained when minimizing the function only for the scale value and not using RANSAC, as done in earlier approaches [HLS14, JT01]. The system setup required by these methods is very specific and incompatible with a fully mobile implementation (e.g., Jung and Taylor [JT01] employ a robotic setup with a high accuracy professional IMU and an omnidirectional camera, and camera motion is estimated on manually matched image features). We thus assume as comparative values for these methods the best values obtained with the software implementation part and the best results declared in their papers. The estimated error with our system (assuming as ground truth the manual measurement) is on average less than 3% while with just scale minimization without RANSAC it is about 27%. In two cases, *Statuettes* and *Office desk*, the results are similar.

We noticed that our approach is not affected by the sensor rate (see case *Workstation Fastest*), since our RANSAC statistic approach compensates for the inevitable presence of additional noise with a greater number of samples. To this purpose an adaptive confidence threshold, $C_{th} = 0.01$ for the Lo-RANSAC step is chosen to keep 80% of inliers couples.

We experienced instead that both our method and the methods of Ham et al. [HLS14] and Jung and Taylor [JT01] are strongly affected by SfM features tracking. In fact, if user motion is too fast, resulting in many lost poses, the quality of the results degrades consequently (case *Desk fast motion* in Table 1). This leads to the fact that our method, similarly to previous approaches, works on the assumption that a sufficient number of features can be tracked within the scene. Unoccluded scenes images with a not exceedingly high motion are thus the best domain of application of our method. On the contrary, scenes with large amounts of occlusions/disocclusions are more challenging.

7. Conclusions

We presented a new approach to the problem of metric scale reconstruction. The method recovers real world units by comparing, through a specialized RANSAC-based algorithm, the acceleration values as obtained by the IMU to those derived from image data. The proposed system extends previous state-of-the-art solutions and has been proved effective on a variety of test cases. Although the algorithm was designed for the specific purpose of acquiring object of average scale, typically indoor furniture, its effective extension to larger objects only requires handling a non-constant set of tracked features and it can be regarded as a pure implementation issue.

In contrast to previous works, our approach needs an acquisition video of just few seconds and can be implemented on common mobile devices equipped with an accelerometer. Since the algorithm was designed to easily run on mobile systems, we also see a natural use in the next generation mobile Virtual Reality devices, such as Samsung Gear 360 and others, for example for rapidly providing metric information about the surrounding environment.

Acknowledgements This work has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 607737 (VASCO). We also acknowledge the contribution of Sardinian Regional Authorities under projects VIGEC and Vis&VideoLab.

References

[APS15] ALIAKBARPOUR H., PALANIAPPAN K., SEETHARAMAN G.: Fast structure from motion for sequential and wide area motion imagery. doi:http://pamitc.org/iccv15/. 2

[Cap05] CAPEL D.: An effective bail-out test for ransac consensus scoring. In *Proc. BMVC* (2005), pp. 629–638. 5

[CCR08] CIGNONI P., CORSINI M., RANZUGLIA G.: Meshlab: an open-source 3d mesh processing system. *ERCIM News*, 73 (April 2008), 45–46. 6

[Cha01] CHAI L.: Three-dimensional motion and structure estimation using inertial sensors and computer vision for augmented reality. *IEEE Transactions on Robotics 11*, 5 (Mar 2001), 474–492. doi:10.1162/105474602320935829. 2

[DSM11] DONG-SI T. C., MOURIKIS A. I.: Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on* (May 2011), pp. 5655–5662. doi:10.1109/ICRA.2011.5980267. 2

Scene Name		Real scale m / s.u.	Acquisition info Seconds	Poses	Samples	Our approach m / s.u.	Error	Simple scaling m / s.u.	Error
3D printer		2.094	17.0	65	883	2.01	4.0%	2.85	36.1%
Scanner setup		3.565	9.8	53	641	3.45	3.1%	3.12	12.4%
Desktop		6.520	11.3	48	596	6.24	4.2%	5.16	20.8%
Statuettes		2.602	11.5	53	607	2.49	4.5%	2.48	4.9%
Office desk		1.977	30.4	88	471	2.01	1.8%	2.01	1.8%
Office workstation		3.95	12.3	37	1307	3.94	0.3%	3.98	0.6%
Ara pacis		1.568	30.07	77	1569	1.52	2.8%	1.80	13.0%
Workstation (Fastest)		0.707	9.9	34	1305	0.73	2.7%	0.89	20.4%
Desk fast motion		6.918	14.8	74	1718	6.28	9.1%	3.88	44.0%

Table 1: ***Scale factor estimation.*** *Comparison vs. ground truth and other approaches. We present for each dataset the real ratio between meters (m) and scene units (s.u.) assumed as ground truth, the duration of the acquisition video, the number of original camera poses as they have been returned by the SfM pipeline and the number of acceleration samples. We indicate our results in column* Our approach*, while column* Simple scaling *indicates the results obtained by minimizing the function only for the scale value without using RANSAC [HLS14, JT01].*

[FCDS15] FORSTER C., CARLONE L., DELLAERT F., SCARAMUZZA D.: Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Proceedings of Robotics: Science and Systems* (Rome, Italy, July 2015). doi:10.15607/RSS.2015.XI.006. 2

[For88] FORNBERG B.: Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation 51*, 184 (1988), 699–699. doi:10.1090/S0025-5718-1988-0935077-0. 5

[FPL*10] FRAHM J.-M., POLLEFEYS M., LAZEBNIK S., GALLUP D., CLIPP B., RAGURAM R., WU C., ZACH C., JOHNSON T.: Fast robust large-scale mapping from video and internet photo collections. *ISPRS Journal of Photogrammetry and Remote Sensing 65* (Nov. 2010), 538–549. doi:10.1016/j.isprsjprs.2010.08.009. 2

[HLS14] HAM C., LUCEY S., SINGH S.: *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV*. Springer International Publishing, 2014, ch. Hand Waving Away Scale. 2, 3, 4, 5, 6, 7

[HR01] HUSTER A., ROCK S. M.: Relative position estimation for intervention-capable auvs by fusing vision and inertial measurements. In *Proceedings of the 12th International Symposium on Unmanned Untethered Submersible Technology* (2001), Durham NH: August. 2

[JEJR04] J. M., E. M., J.S. B., R. M.: *Manual of Photogrammetry, fith edition*. American Society of Photogrammetry and Remote Sensing, 2004. 2

[JS11] JONES E. S., SOATTO S.: Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. J. Rob. Res. 30*, 4 (Apr. 2011), 407–430. doi:10.1177/0278364910388963. 2

[JT01] JUNG S.-H., TAYLOR C.: Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 2, pp. II–732–II–737 vol.2. 2, 3, 4, 6, 7

[Kal60] KALMAN R. E.: A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D (1960), 35–45. 2

[Lhu12] LHUILLIER M.: Incremental fusion of structure-from-motion and gps using constrained bundle adjustments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 12 (Dec 2012), 2489–2495. doi:10.1109/TPAMI.2012.157. 2

[LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 1981), IJCAI'81, Morgan Kaufmann Publishers Inc., pp. 674–679. URL: http://dl.acm.org/citation.cfm?id=1623264.1623280. 3

[LLB*14] LEUTENEGGER S., LYNEN S., BOSSE M., SIEGWART R., FURGALE P.: Keyframe-based visualâĂŞinertial odometry using nonlinear optimization. *The International Journal of Robotics Research* (2014). doi:10.1177/0278364914554813. 2

[LS08] LUPTON T., SUKKARIEH S.: Removing scale biases and ambiguity from 6dof monocular slam using inertial. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* (May 2008), pp. 3698–3703. doi:10.1109/ROBOT.2008.4543778. 2

[Mar12] MARTINELLI A.: Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics 28*, 1 (Feb 2012), 44–60. doi:10.1109/TRO.2011.2160468. 2

[Ope15] Opencv, open source computer vision library. http://opencv.org/platforms/android.html, 2015. Online; version 3.0. 6

[PLST07] PINIES P., LUPTON T., SUKKARIEH S., TARDOS J. D.: Inertial aiding of inverse depth slam using a monocular camera. In *Robotics and Automation, 2007 IEEE International Conference on* (April 2007), pp. 2797–2802. doi:10.1109/ROBOT.2007.363895. 2

[PRCZ12] PORZI L., RICCI E., CIARFUGLIA T. A., ZANIN M.: Visual-inertial tracking on android for augmented reality applications. In *Environmental Energy and Structural Monitoring Systems (EESMS), 2012 IEEE Workshop on* (Sept 2012), pp. 35–41. doi:10.1109/EESMS.2012.6348402. 2

[SFF14] SCHÖNBERGER J. L., FRAUNDORFER F., FRAHM J.-M.: Structure-from-motion for mav image sequence analysis with photogrammetric applications. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 1* (2014), 305–312. 2

[ST94] SHI J., TOMASI C.: Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)* (1994), pp. 593 – 600. 3

[TGL*10] TARDIF J. P., GEORGE M., LAVERNE M., KELLY A., STENTZ A.: A new approach to vision-aided inertial navigation. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on* (Oct 2010), pp. 4161–4168. doi:10.1109/IROS.2010.5651059. 2

[TKM*13] TANSKANEN P., KOLEV K., MEIER L., CAMPOSECO F., SAURER O., POLLEFEYS M.: Live metric 3d reconstruction on mobile phones. In *Computer Vision (ICCV), 2013 IEEE International Conference on* (Dec 2013), pp. 65–72. doi:10.1109/ICCV.2013.15. 2, 3, 4

[TZ00] TORR P., ZISSERMAN A.: Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding 78*, 1 (2000), 138 – 156. doi:http://dx.doi.org/10.1006/cviu.1999.0832. 5

[Ume91] UMEYAMA S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 4 (Apr. 1991), 376–380. 4

[WS11] WEISS S., SIEGWART R.: Real-time metric state estimation for modular vision-inertial systems. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)* (2011). 2