Available online at www.sciencedirect.com

ScienceDirect

Image and Vision Computing 00 (2016) 1–16

Procedia Computer Science

True Scaled 6 DoF Egocentric Localisation with Monocular Wearable Systems

Daniel Gutierrez-Gomez*, J.J. Guerrero*

Instituto de Investigacion en Ingenieria de Aragon (I3A) and Departamento de Informatica e Ingenieria de Sistemas (DIIS), Universidad de Zaragoza, 50018 Zaragoza, Spain

Abstract

In this work we present a novel approach to obtain scaled odometry and map estimates when performing monocular SLAM with wearable cameras. After proving first that the oscillation of the body during walking can be observed in the odometric estimate from a monocular SLAM algorithm, we develop a method to estimate the walking speed from the frequency of this oscillation. Having the real walking speed, a scale factor can be dynamically computed to obtain a true scaled estimate of the map and visual odometry, avoiding scale drift on long term trajectories. Although the algorithm requires the person to be walking in order to estimate the scale, the experiments, carried out in outdoor and indoor environments and with different types of cameras, show that our method is reliable and robust to challenging situations like stops, changes in pace or stairs, and provides a significant improvement with respect to the initial unscaled estimate. It also outperforms state-of-the-art solutions to correct the scale drift in monocular SLAM, giving in addition the absolute scale of the trajectory and the 3D observed scene.

Keywords: Monocular SLAM, wearable vision, egocentric localisation.

1. Introduction

The use of cameras on sensing platforms arouses great interest due to the low cost of this kind of sensors and the high amount of information which is encoded in one image. In addition to this, the improvement on CPU performance and the advances in camera miniaturization and mobile computing have lead to an emerging interest in the use of wearable cameras [1], which are now available as consumer products (Memoto, GoPro). Besides recreational use, wearable cameras can also provide assistance to impaired people [2], [3].

In the context of a wearable vision system, a precise odometric localisation of the camera and representation of the environment, can be used as a first step to provide high quality semantic information. The use of cameras to tackle the problem of Simultaneous Localisation and Mapping (visual SLAM) has extended over the last years. Traditionally, the visual SLAM problem has been addressed from a bayesian filtering perspective for example using an Extended Kalman Filter [4] or particle filters [5]. On the other hand, structure from motion (SfM) approaches [6], [7], [8], which address the visual SLAM problem from an optimisation perspective, have recently emerged as an alternative to the probabilistic filters due to their higher precision and the ability of building denser maps in real-time.

However, due to their purely projective nature, monocular vision systems are not able to provide depth measurements of the observed landmarks. One would need two images, provided that the baseline between the corresponding camera poses is large enough, to estimate both the camera translation and the depth of the observed landmarks up to an ambiguity in the scale factor.

In the context of monocular SLAM, this limitation translates in two closely related problems. The first one is a chicken-egg problem during initialisation by which neither camera translation nor the depth of the landmarks can be estimated until enough parallax is observed. This can be tackled by providing some initial landmarks whose 3D position is fully known [4], by user-aided initialisation [6], or by using an appropriate parametrisation which allows for an automatic initialisation of the monocular SLAM [9].

The second one is known as scale drift. During the initialisation of monocular SLAM, the scale is initially fixed to a real or an arbitrary value, depending of the used method of the mentioned before. However, far from keeping itself constant, the continuous lost of old landmarks and the initialisation of new

*Corresponding author

Email addresses: danielgg@unizar.es (Daniel Gutierrez-Gomez), josechu.guerrero@unizar.es (J.J. Guerrero) ones gives raise to a change of the scale of the scene as the camera moves. This ends up by introducing a deformation in the final trajectory and map estimates which goes beyond a simple scale ambiguity. The deformation introduced by the drift in the scale can even persist after applying state-of-the-art loop closure techniques.

In this paper we propose a novel approach to compute dynamically the true scale from visual odometry estimates obtained with wearable single cameras (Fig. 1), avoiding scale drift problems in large environments. Our method is specially suited to be used in wearable systems since it takes advantage of the characteristic oscillatory movement of human body during walking to extract the scale information. The implementation is done within a state-of-the-art visual SLAM approach [10]. Nevertheless as it only needs the output corresponding to the trajectory estimate it can be used in any visual odometry system, provided that the accuracy and the time resolution of the SLAM algorithm are fine enough to capture the camera oscillation associated to walking. The successful results of our approach within an EKF-SLAM system means that it should provide also good results with the more recent and more accurate visual SLAM approaches based on Bundle Adjustment [6], [11]. Also, to clearly perceive the walking oscillations some restrictions must be fulfilled. Firstly, the user must be walking on a relatively plain terrain, such that its roughness is lower than the amplitude of the walking oscillations. Secondly, the camera must be attached to a body part whose motion is mainly due to the action of walking, i.e., our method is quite likely to fail if camera is attached to one arm of the user.

The method works as follows (Fig. 2): given an initial unscaled section of the trajectory composed by N camera poses, we extract the signal of the vertical component of the camera position and estimate the step frequency by computing its Discrete Fourier Transform (DFT). The real walking speed is computed from the step frequency using a biomedical relation [12] which is supported by the natural tendency to optimise the metabolic cost of walking [13, 14]. This speed measurement is fed into a computationally simple particle filter to compute a dynamic scale factor to scale each trajectory section.

Additionally, our method is made robust against bad scale factor estimates in two ways. First, the spectral power of the candidate step frequency is tested to be consistent with the amplitude of the walking oscillations. This test makes possible the detection of non-walking situations, e.g., when the user is stopped, and then allowing for a different update strategy of the scale factor. A second compatibility test is performed during particle filtering to reject big variations of the scale factor.

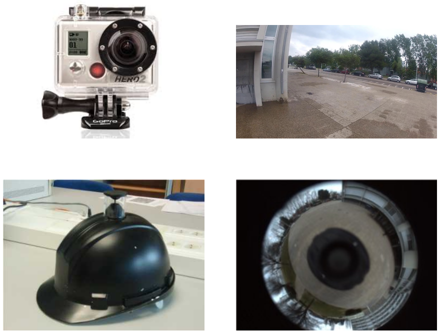


Figure 1. Devices used in our experiments. On top, GoPro Hero 2 wide angle camera. On bottom, our helmet with catadioptric camera consisting on a conic mirror and a VS-C14U-80-ST catadioptric camera.

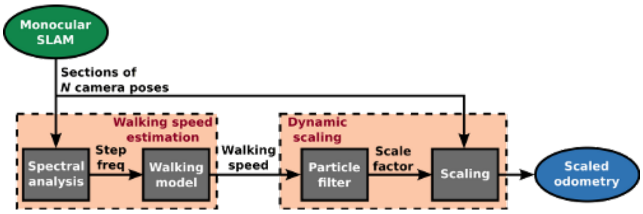


Figure 2. Scheme of the basic scaling method.

Part of this work was presented as conference papers in [15] and [16]. Now in this paper all the previous conference results obtained by experimentation with a catadioptric camera are integrated and improved. Also, the paper includes the following novel contributions:

- A method to scale not only the visual odometry but also the detected points of the scene.
- New experiments with GoPro-like wearable cameras showing the suitability of our method also for its use with more universalised cameras.
- Comparison of our approach to the proposal by Strasdat et al. [8] for scale drift removal.
- Evaluation of sensitivity of our method to inaccurate fitting of the user specific parameters of the empirical formula relating frequency to speed.
- Testing of our method when the camera is worn on the chest rather than on the head.

The rest of the paper is organised as follows. Section 2 discusses the related work on wearable vision and on the scale estimation problem in monocular SLAM. Section 3 describes the SLAM approach used in this paper. Section 4 describes how the real walking speed is approximated from the unscaled odometry estimate. Section 5 explains the computation of the scale and the subsequent scaling of the trajectory and features. Section 6 demonstrates the effectiveness of our approach using both omnidirectional and conventional monocular systems. Conclusions and final discussion are presented in Section 7.

2. Related work

The main topics for related work in this paper are wearable vision and the scale problem in monocular SLAM.

2.1. Wearable vision

The research on wearable cameras for personal aiding has widespread since the pioneering works of Mann [1]. Wearable cameras are normally placed in locations like the head, shoulders or chest, which, for a general purpose, allow for a wide field of view and resilience to body motion [17].

Most works on wearable vision systems have developed towards recognition of human activities, where many approaches take advantage from a nice feature of chest-wearable cameras: the prior knowledge of the action or manipulation taking place at the centre of the image. Recognition problems where wearable systems have been applied are segmentation of handled objects [18, 19], recognition of activities and objects in the workspace [20], novelty detection in a daily routine [21], clustering of sport activities from video sequences [22], human detection [23] or analysis of human movements to infer social interactions [24].

With respect to the use of wearable vision for localisation, some works [25, 26] propose appearance based localisation methods in indoor environments. Concerning odometric localisation, Mayol-Cuevas et al. [27] presented a wearable active vision system which changes its heading direction and uses monocular SLAM for self-localisation. A similar system is presented by Castle et al. [28], using monocular SLAM and object recognition for augmented reality. In [29] Badino and Kanade propose head-wearable stereo system to estimate structure and motion. Alcantarilla et al. [30] propose a wearable stereo system which computes, together with SLAM, an estimate of the dense scene flow to segment moving objects in the scene. Also some works [31, 32] propose wearable platforms for human localisation and navigation without vision sensors, combining an Inertial Measurement Unit (IMU) and a laser scanner.

2.2. Monocular SLAM and the scale problem

The problem of estimating the true scale in monocular SLAM is addressed either by using additional proprioceptive sensors, like IMUs and odometers, which provide metric information; or by considering geometric priors or constraints, mainly in robotic platforms.

Among solutions using additional sensors, Lupton and Sukkarieh [33] make the true map scale observable by integrating the visual data and the IMU data within an information filter. Nützi et al. [34] propose an approach where the scale is computed by fusing the position and velocity from the visual odometry estimate with the IMU data in an Extended Kalman Filter (EKF). In [35], Engel et al. compute the scale factor of a quadricopter visual odometry estimate from the measurements of its on-board IMU and altimeter by using an optimisation scheme.

Cumani et al. [36] use the wheel odometry to provide a prior estimation of the true scaled motion between two consecutive frames which is refined by the update from camera measurements. Eude et al. [37], take the odometric measurement of distance between two camera poses to compute a scale factor which is applied to the displacement estimated with the camera. Similarly, Scaramuzza et al. [38] use the vehicle speed measurement to compute the true distance between the last two frames and recover the 3D structure by triangulation of the common image points. In [10], Civera et al. also obtain scaled visual odometry estimates by fusing the wheel odometry and visual information in a EKF-SLAM framework.

Concerning works not using additional sensors, Lothe et al. [39] use the prior knowledge of the distance from the camera to the ground plane, which they obtain from the estimated 3D points of the scene, to compute the scale factor of the scene. Song and Chandraker [40] also use the ground plane for scale estimation, but in addition to 3D points they also use dense alignment of ground plane templates and cues from detected objets to estimate the scale factor. Scaramuzza et al. [41] obtain directly the scaled estimate of a vehicle trajectory by tracking only the points and using an homoghraphy-based ground plane navigation. In [42] Scaramuzza et al. exploit non-holonomic motion constraints of wheeled vehicles to resolve the scale when the vehicle turns. Botterill et al. [43] propose a solution to the scale drift problem consisting in identifying previously learnt object classes of the environment and measuring the size of these objects to improve the scale estimate. Strasdat et al. [8] propose a loop closure technique in which camera poses are defined as 7 DoF similarity transformations (translation, rotation and scale) rather than the habitual 6 DoF rigid transformations (translation and rotation). This allows to correct the deformation produced by the scale drift, though it still remains a scale ambiguity in the final estimate. In [44], Hansen et al. develop a monocular visual odometry system for localisation in gas pipes, where the scale is estimated by fixing patterns of known size in the interior walls of the pipe.

The method proposed in this paper fits in the second category, since we get the scale only using the visual information and a prior given by a walking model which depends on the height and specific user parameters. The strength of our method is that it is explicitly thought to operate on human wearable systems, where unlike to ground vehicles, odometric information from encoders is not available to recover the scale.

On the other hand, faced to scale estimation methods not using odometric measurements, all of them have their own drawbacks. Using the IMU, besides involving an additional sensor, to obtain scale information we need to know the initial speed, which, if not known *a priori*, can only be estimated by performing a joint IMU-camera initialisation. The estimation of the scale by the identification of the ground plane can be problematic in environments with a poor textured ground and containing other dominant planes like walls. The identification of known objects of the scene relies in the detection of a set of previously learnt object categories which could be affected by false positives. The correction of the scale drift at loop closures is only possible when one region of the environment is revisited.

Thus, given that all the approaches have their own limitations, an alternative way to compute the scale as proposed in this paper has always a beneficial effect.

3. Monocular SLAM

Monocular SLAM algorithms aim to estimate a visual odometry and at the same time build a map of landmarks using only the measurements from a single camera. The algorithm we use in this work is based in the monocular EKF-SLAM proposed in [10] by Civera et al. and its adaptation to omnidirectional cameras [45]. The camera state as well as the position of the landmarks at time step i are encapsulated in a state vector $\mathbf{x}_i$

$$\mathbf{x}_i = (\underbrace{\mathbf{r}^C_{W,i},\ {}_W\mathbf{q}^C_i,\ \mathbf{v}^C_{W,i},\ \omega^C_{C,i}}_{\text{Camera state } \mathbf{x}^C_i},\ \underbrace{\mathbf{r}^{C(j)}_{W,i}, \theta^{(j)}_i, \phi^{(j)}_i, \rho^{(j)}_i, ...}_{\text{3D points (IDP) } \mathbf{y}^{(j)}_W}), \tag{1}$$

where $\mathbf{r}^C_{W,i}$ is the camera position, ${}_W\mathbf{q}^C_i$ is the quaternion of its orientation and $\mathbf{v}^C_{W,i}$ and $\omega^C_{C,i}$ are its linear and angular velocities, respectively.

The 3D locations are parametrised in an anchored inverse depth parametrisation (IDP) [9], where $\mathbf{r}^{C(j)}_{W,i}$ is the anchor camera position where the landmark was first viewed, $\theta^{(j)}_i$, $\phi^{(j)}_i$ and $\rho^{(j)}_i$ are respectively the elevation angle, the azimuth angle and the inverse depth with respect to the anchor position.

In an EKF based SLAM a motion and a measurement model must be provided. The motion model describes the change on the camera pose from time step $i-1$ to i and it is described by the following equation:

$$\mathbf{x}^C_i = \mathbf{f}(\mathbf{x}^C_{i-1}, \mathbf{u}_i) + \mathbf{w}_i, \tag{2}$$

where $\mathbf{f}(\cdot)$ is the state transition function , $\mathbf{x}^C_{i-1}$ is the past camera pose, $\mathbf{u}_i$ is the control input and $\mathbf{w}_i$ is a Gaussian noise with covariance $\mathbf{Q}_i$. This model is used to propagate the state estimate $\mathbf{x}_i$ and its covariance $\mathbf{P}_i$ in the EKF prediction step. Internal measurements from an odometer or an IMU can be integrated as control inputs $\mathbf{u}_i$. Alternatively it is often used a constant velocity model ($\mathbf{u}_i = \mathbf{0}$) where possible changes in velocity are taken care of by introducing large acceleration noise priors as the process noise $\mathbf{w}_i$.

The measurement model is used to introduce the information from measurements of external sensors in the EKF update step. In monocular SLAM it is defined by an observation function which encapsulates a projectivity transformation. The observation function depends on the characteristics of the vision system. For central projection systems, i.e., planar (conventional), dioptric or catadioptric systems, it can be divided in these two steps: one projection onto a unit sphere independent from camera parameters and a non-linear mapping from the sphere to the image plane which accounts for the camera geometry and calibration parameters [46]. This is synthesised in the following equations:

$$\mathbf{z}^{(j)}_i = \mathbf{h}\left(\mathbf{x}^C_i, \mathbf{y}^{(j)}_W\right) = \mathbf{K}_c \hbar \left(\xi, \frac{\mathbf{y}^{(j)}_{C,i}}{\left\|\mathbf{y}^{(j)}_{C,i}\right\|}\right) \tag{3}$$

$$\mathbf{y}^{(j)}_{C,i} = \mathbf{R}^{\mathsf{T}}\left({}_W\mathbf{q}^C_i\right)\left(\mathbf{r}^{C(j)}_{W,i} - \mathbf{r}^C_{W,i} + \frac{1}{\rho^{(j)}_i}\mathbf{m}\left(\phi^{(j)}_i, \theta^{(j)}_i\right)\right), \tag{4}$$

where ξ is a parameter encapsulating the geometric properties of the camera, $\hbar$ is a non-linear function, $\mathbf{K}_c$ is a conventional camera calibration matrix, and $\mathbf{m}(\cdot)$ the function which maps the elevation and azimuth angles to a unit vector.

With this generalised model, the observations of the tracked features (those in the EKF state vector) are predicted, and then putative matches $\mathbf{z}^{(j)}_i$ are obtained by active search in the uncertainty region given by the bound of the 95% confidence interval of the projection $\mathbf{S}^{(j)}_i$ of the state covariance:

$$\mathbf{S}^{(j)}_i = \mathbf{H}^{(j)}_i \mathbf{P}_i \mathbf{H}^{(j)^{\mathsf{T}}}_i + \mathbf{R_n}, \tag{5}$$

where $\mathbf{H}^{(j)}_i$ is the jacobian of the measurement function and $\mathbf{R_n}$ the measurement noise in the image.

Spurious matches are rejected by RANSAC, and then correct matches are used to update the state both of the camera and the landmarks.

3.1. Map Management

To keep the computational cost low, only a few features, referred to as *tracked features* $\mathcal{T}_i$, are kept within the EKF state. Tracked features are divided in *point* (low depth uncertainty) and *ray* (high depth uncertainty) features. At each step, features which failed to be matched in the last frames are removed from the EKF. The set of removed features at step i is denoted by $\mathcal{E}_i = \left[\mathcal{E}^r_i, \mathcal{E}^p_i\right]$

The removed point features are added to an independent fixed map $\mathcal{M}_{E,i}$ which is not updated by the EKF:

$$\mathcal{M}_{E,i} = \left[\mathcal{M}_{E,i-1}, \mathcal{E}^p_i\right], \tag{6}$$

being the global map composed by the point tracked features and the features in $\mathcal{M}_{E,i}$

$$\mathcal{M}_i = \left[\mathcal{T}^p_i, \mathcal{M}_{E,i}\right]. \tag{7}$$

3.2. Loop closure

To make possible a fair evaluation of our method using the ground truth trajectory, we correct the noticeable odometry drift which inevitably arises in large experiments by closing the loop at revisited areas. Closing the loops makes also possible the comparison of our proposal with a state-of-the-art method to correct the scale drift [8].

To compute the relative pose between loop frames we use the libraries OpenCV [47] for feature extraction and matching and OpenGV [48] for geometric algorithms. Given a loop detected between a recent frame j and an older frame i , we must estimate the rigid body motion between them expressed by the transformation:

$${}_j\mathbf{T}^i = \begin{pmatrix} {}_j\mathbf{R}^i & \mathbf{r}^i_j \\ 0 & 1 \end{pmatrix} \in SE(3). \tag{8}$$

To do so, first, from 2D-2D correspondences, we robustly estimate with RANSAC the relative pose between frame j and a close frame j_{aux} with enough parallax, rescaling the translational part by taking the norm of the translation between frames from the visual odometry. Secondly, correct correspondences between frames j and j_{aux} are triangulated, and finally ${}_j\mathbf{T}^i$ is obtained from 3D-2D correspondences between the triangulated

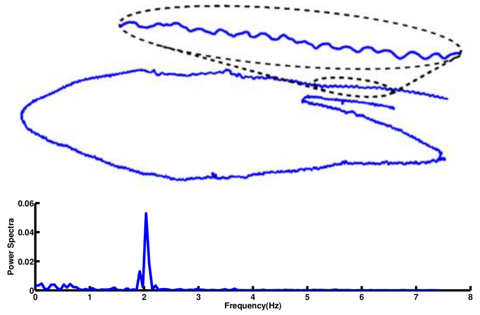


Figure 3. Top: Trajectory estimate of Visual SLAM from a head-mounted catadioptric camera including a partial zoom. Bottom: Power spectra of the vertical component

3D points referenced at frame j and the keypoints extracted in frame i.

For the comparison with [8] we need to use similarity transforms:

$$ {}_j\mathbf{S}^i = \begin{pmatrix} {}_js^i\,{}_j\mathbf{R}^i & \mathbf{r}^i_j \\ 0 & 1 \end{pmatrix} \in Sim(3), \tag{9}$$

where we require an additional parameter ${}_js^i$ to compute the loop constraint. To estimate it, first we compute ${}_i\mathbf{T}^j$ in the same way as with ${}_j\mathbf{T}^i$ by interchanging the roles of i and j. Note that in presence of scale drift ${}_j\mathbf{T}^i\,{}_i\mathbf{T}^j \neq \mathbf{I}$ since translation of each transform has a different scale propagated from the visual odometry at the frame taken as reference for the triangulated 3D points. However, introducing the extra scale parameter in the transform we can apply ${}_j\mathbf{S}^i\,{}_i\mathbf{S}^j = \mathbf{I}$ and thus we get:

$$ {}_js^i\,{}_j\mathbf{R}^i\mathbf{r}^j_i + \mathbf{r}^i_j = \mathbf{0} \implies {}_js^i = \frac{\left\|\mathbf{r}^i_j\right\|}{\left\|\mathbf{r}^j_i\right\|}. \tag{10}$$

4. Walking speed estimation

Our scheme for the estimation of the real walking speed is based in two hypotheses. First, the oscillation of the body during walking can be observed in the visual odometry (Fig. 3). The second hypothesis is the existence of a tight correlation between the step frequency and the stride length which allows to estimate the walking speed without knowing the later.

4.1. Walking speed - step frequency relation

The estimation of the walking speed is based on its close correlation with the step frequency, which was shown in the work by Grieve et al. [12]. This correlation is encapsulated in a power law where the walking speed (V_{walk}) normalised with height (H) is presented as a function of the step frequency (f_{step}):

$$V_{walk} = \alpha f_{step}^{\beta} H, \tag{11}$$

where V_{walk} is in m/s, f_{step} in Hz, H in m, and α and β are characteristic parameters which differ from one individual to another. The mean values provided in [12] for these parameters are, after converting to I.S. units, $\alpha = 0.2896$ and $\beta = 1.7544$. Noting that:

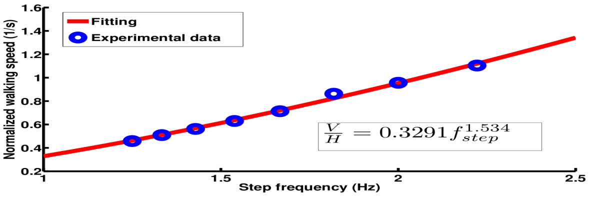


Figure 4. Power fitting of the experimental data to compute the relation between walking speed and step frequency ($\mu_{err} = 0.018$, $max_{err} = 0.04$).

$$V_{walk} = f_{step} L_{stride}, \tag{12}$$

and substituting in (11), it can be observed, that though not explicitly shown, our method in essence is taking the stride length L_{stride} as the geometric prior needed to get the absolute scale, and computes it as a function of the step frequency and specific user parameters (α, β and height).

By measuring the oxygen consumptions of different subjects under forced and free gaits for a set of speeds, Zarrugh et al. [13] showed that the relation in (11) is the result of a human tendency to choose a step frequency which minimises the metabolic cost of walking. Kuo [14] proposed a metabolic cost function modelling the combined actions of pushing off and swinging the leg, whose minimisation predicts the preferred relationship presented in [12].

Although walking speed can be estimated using the mean values provided in [12], for higher accuracy we use our own α and β parameters explicitly computed for the camera operator. We measured the time t_i it took the operator to walk a distance $s = 100$ m at the times per step ΔT_i given by a metronome ranging from 0.45 to 0.80 seconds in intervals of 0.05 seconds. The metronome emitted a beep at the time intervals it was set up in order to force the user to synchronise his pace. The height of the operator is $H = 1.88$ m. During the experimentation, it was noticed that measurements above and below the considered interval of times per step were not possible due to the impossibility to consistently synchronise with the metronome beat. Thus, we have established a range of feasible step frequencies between $f_{st}^- = 1$ Hz and $f_{st}^+ = 3$ Hz. Normalised walking speeds V_i' and step frequencies f_i were computed from the raw experimental data. Then a power fitting was applied to obtain the values of $\alpha = 0.329$ and $\beta = 1.534$ (Fig. 4).

4.2. Estimation of the step frequency

Given the previous biomedical relation, the problem of estimating the walking speed, can be translated into estimating the step frequency.

To correct scale drift, the walking speed must be computed periodically in sections or windows of N camera poses. If we let i be the SLAM time step index, each camera pose $\mathbf{x}_i^C$ can be equivalently notated as $\mathbf{x}_k^C(n)$ assigning a section k and an index n within that section as follows:

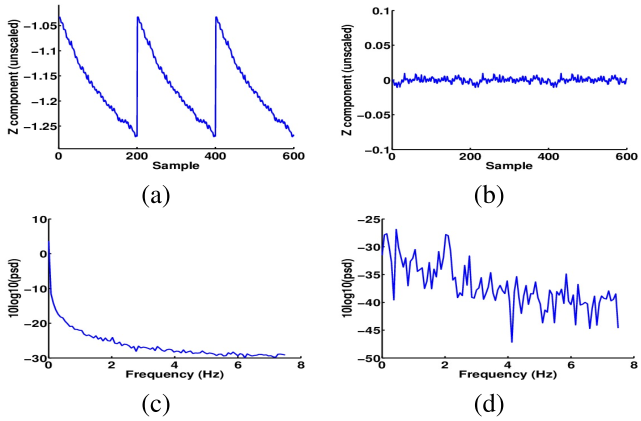


Figure 5. Z-component signal segment (top) and corresponding power spectra in logarithmic scale (bottom) of two instances from the same visual odometry section: (a,c) without preprocessing the input signal and (b,d) with offset elimination and filtering of the input signal. Note how in (b) the power peak at the step frequency (2 Hz) is observable and the highest in the interval of feasible step frequencies. Signal segments have been copied three times to make visible the difference in the discontinuity between the two instances.

$$k = \mathrm{Int}\left(\frac{i-1}{N}\right), \tag{13}$$

$$n = i - (k-1)N. \tag{14}$$

The world reference frame for the visual odometry is fixed by the initial camera pose. In the experimental setups for the different cameras, the camera frame is oriented so that one of its axis is approximately aligned with the normal to the ground plane. This is a reasonable assumption for a camera worn on the head or some part of the trunk. Also, since a fixed transformation known *a priori* can be applied so that the z-axis of the world frame is aligned with the ground normal, without loss of generality we assume that the body oscillation is observed in the z-component of the trajectory.

The step frequency is estimated by extracting the spectral composition of the oscillatory component of the trajectory estimate using the DFT. When computing the Fourier Transform of discrete signals, the gap between the first and the last elements of the signal, the variations or the drift of the ground plane, and the slight miss-alignment of the reference z-axis with the ground normal can introduce low frequency harmonics. To counteract these effects the data sequence is preprocessed by subtracting the first element $z_k(1)$ from all the elements and applying a second order high pass digital filter with a cut-off frequency of $f_c = 0.7$ Hz, which provides a good attenuation of the low frequency harmonics, without affecting the harmonics in the range of feasible step frequencies. (Fig. 5).

After filtering the data the spectral composition is obtained by the DFT:

$$\Gamma_{d,k}(f_m) = \frac{1}{F_s N}\left\|\sum_{n=1}^{N} z_k(n)\exp\left(-j\frac{2\pi f_m(n-1)}{F_s}\right)\right\|^2 \tag{15}$$

$$f_m = \frac{mF_s}{N} \qquad m = -\frac{N}{2},...,-1,0,1,...,\frac{N}{2}, \tag{16}$$

where Γ_d power spectral density function, F_s is the sampling frequency, which in our case is the number of frames per second (fps) of the camera, and f_m are the frequencies for which the spectrogram is sampled.

The sampling frequency must be high enough to avoid aliasing when the maximum feasible step frequency f_{st}^+ occurs. Also, the number of poses taken must be high enough to provide a good resolution of the spectrogram, with its lower limit given by the minimum number of samples needed to observe at least one oscillation in the case of the minimum feasible step frequency f_{st}^- .

For the computation of the DFT N has to be as high as possible, but from a global point of view, a high N involve a less frequent update of the scale factor and a reduced ability to detect changes in the walking speed. This can result in a decreasing accuracy in the scaled trajectory estimate. Moreover, if interested in real time operation, the time delay to update the scaled trajectory grows linearly with N, since before scaling one section we need to get the new N unscaled camera poses from the SLAM algorithm. Thus, in summary, for the choice of N we must reach a trade-off between the accuracy of the DFT and the frequency with which the scale factor is updated.

Given the spectrogram $\Gamma_{d,k}(f_m)$, the step frequency $f_{st,k}$ is estimated as:

$$f_{st,k} = \operatorname*{arg\,max}_{f_m \in [f_{st}^-, f_{st}^+]} \Gamma_{d,k}(f_m). \tag{17}$$

4.3. Detection of non-walking situations

At this point the method to estimate the step frequency would return an estimate regardless of whether the user is walking or not. To discard erroneous estimates when the user is not walking we check that the spectral power $\bar{P}(f_{st,k})$ of the computed step frequency to be consistent with a range of feasible oscillation amplitudes during walking bounded by a A_z^+ and A_z^- . Applying the Parseval's theorem, which states that the energy of a signal is preserved in the frequency domain, we can approximate the energy $\bar{P}$ of the signal corresponding to the body oscillation as:

$$\bar{P}(f_{st,k}) = 2\frac{F_s}{N}\sum_{m=m^-}^{m^+} \Gamma_d(f_{m,k}), \tag{18}$$

with $m^- = \mathrm{round}\left(N\frac{f_{st}-\Delta f}{F_s}\right)$, $m^+ = \mathrm{round}\left(N\frac{f_{st}+\Delta f}{F_s}\right)$, and where $f_{st,k}$ is the estimated step frequency, Γ_d is the discretised spectrogram of the z-component of the camera poses, F_s is the sampling frequency of the camera, N the camera poses in the analysed section and Δf the frequency interval centred at $f_{st,k}$ along which the energy is spread along.

Since the power spectral density is computed for the unscaled z-component of the visual odometry, the computed power must be scaled by multiplying it by the square of the current mean scale factor $\bar{d}_k$. Thus, assuming that the body oscillation is sinusoidal, the condition for the spectral power consistency of the step frequency yields:

$$\frac{1}{2}A_z^{-2} \leq \bar{d}_k^2\bar{P}(f_{st,k}) \leq \frac{1}{2}A_z^{+2}. \tag{19}$$

If this condition is not fullfilled we should choose another strategy. For example if $d_k^2\bar{P}(f_{st,k}) \leq \frac{1}{2}A_z^{-2}$ we may assume that the person is stopped and then avoid updating the scale factor.

5. Dynamic scale update

Having a walking speed estimate and assuming that it is affected by Gaussian noise, the maximum likelihood scale factor for section k could be straightforwardly computed by $d_k = \frac{V_{walk,k}}{\mu_{V,k}}$, where $\mu_{V,k}$ is the average dimensionless speed of the camera poses in section k. However, given the empirical method for the walking speed estimation and the possible variability of the SLAM velocity along N frames, we propose to use a probabilistic filter for the computation of the scale factor. The purpose is to provide robustness against spurious estimations of the walking speed.

After computing the scale factor, the poses and the features measured in current section are scaled using a recursive approach to take care of the scale drift with respect to previous sections.

5.1. Particle Filter for scale factor tracking

For the design of the probabilistic filter we consider a dynamic system whose state $\mathbf{s}_k$ is composed by the magnitude of the SLAM velocity $V_{SLAM,k}$ and the decimal logarithm of the scale factor $\lambda_k = \log_{10}(d_k)$.

$$\mathbf{s}_k = \begin{bmatrix} V_{SLAM,k} \\ \lambda_k \end{bmatrix}. \tag{20}$$

Taking the logarithm has the advantage to naturally restrict the scale factor to positive values allowing to model its uncertainty and noise with additive Gaussian distributions. As a consequence all the non-linearities of the model will be encapsulated in the measurement function.

To track the scale factor, a particle filter with Sampling Importance Resampling is designed [49]. The use of the particle filter is encouraged over an EKF due to its ability to deal with high uncertainty priors of the scale factor which would involve a large linearisation error of the measurement function. Also, since the system state is composed only by 2 variables, the major drawback of the exponential growth of the number of required particles with the number of state variables is negligible.

Hence the state of the system in each section k is approximated by a set of particles:

$$\mathcal{S}_k = \left\{(\mathbf{s}_k^{(L)}, \omega_k^{(L)}) \mid L = 1, 2, ..., P\right\}, \tag{21}$$

where P is the number of particles and $\mathbf{s}_k^{(L)}$ and $\omega_k^{(L)}$ are respectively the state vector and the resampling weight of particle L.

The particles are initialised such that the initial values of $\lambda_0^{(L)}$ are drawn from a Gaussian distribution $\lambda_0 \sim \mathcal{N}(0, \sigma_0)$, where σ_0 is a parameter related to the orders of magnitude in the scale being scoped out.

In the predictive part of the particle filter, particles are sampled down by a proposal distribution $p(\mathbf{s}_k|\mathbf{s}_{k-1})$:

$$\mathbf{s}_k^{(L)} \sim p(\mathbf{s}_k|\mathbf{s}_{k-1}^{(L)}). \tag{22}$$

In our system the sampling of the proposal distribution includes both the update of the non-dimensional speed, which is taken as a control input coming from the visual odometry, and the possible drift in the scale. This is encoded in the following equations:

$$V_{SLAM,k}^{(L)} = \mu_{V,k} + n_v^{(L)} \tag{23}$$

$$\lambda_k^{(L)} = \lambda_{k-1}^{(L)} + n_\lambda^{(L)}, \tag{24}$$

with $n_v^{(L)} \sim \mathcal{N}(0, \sigma_{V,k})$ and $n_\lambda^{(L)} \sim \mathcal{N}(0, \sigma_{drift})$, and where $\mu_{V,k}$ and $\sigma_{V,k}$ are the averaged speed and the corresponding standard deviation of the last set of N SLAM camera poses used for spectral analysis, and σ_{drift} is the standard deviation prior of the scale drift between two consecutive sections, which is modelled as Gaussian noise.

To incorporate the walking speed information into the state estimate, first we need to define a measurement function $h_V(\mathbf{s}_k^{(L)})$ which predicts the walking speed measurement for each particle L.

$$h_V(\mathbf{s}_k^{(L)}) = V_{SLAM,k}^{(L)} 10^{\lambda_k^{(L)}}. \tag{25}$$

To prevent from later accepting spurious estimates of the walking speed, first we take the 95% confidence interval of the histogram of the predicted measurements by eliminating the samples at the tails, thus reducing the number of particles from P to P'.

The remaining samples are weighted by a probability density function $p(V_{walk,k}|\ \mathbf{s}_k^{(L)})$, which captures the statistics of the speed estimate. Assuming that it is affected by Gaussian noise of zero mean and a standard deviation σ_{Vwalk} to be set up empirically, weights are computed as:

$$\omega_k^{(L)} = p(V_{walk,k}|\ \mathbf{s}_k^{(L)}) = \phi\left(\frac{V_{walk,k} - h_V(\mathbf{s}_k^{(L)})}{\sigma_{Vwalk}}\right), \tag{26}$$

where $\phi(z)$ is the probability density function of the standard normal distribution.

Then a consistency test is carried out to verify that the weighted particles lie in the 95% confidence interval of the probability distribution of the measurement model, i.e.:

$$0.95 = P\left(-1.96 \leq \frac{V_{walk,k} - h_V(\mathbf{s}_k^{(L)})}{\sigma_{Vwalk}} \leq 1.96\right). \tag{27}$$

If every particle fails this test, the particle filter iteration ends here. Otherwise the weights of the particles are normalised:

$$\hat{\omega}_k^{(L)} = \frac{\omega_k^{(L)}}{\sum_{p=1}^{P'} \omega_k^{(p)}}, \tag{28}$$

and the set of particles $\mathcal{S}_k$ is resampled by drawing P particles from a multinomial distribution $Mult(P, \hat{\omega}^{(1)}, ..., \hat{\omega}^{(P')})$.

Then for a given section k, the scale factor is obtained by computing the geometric mean of the scale values in the particle set $\mathcal{S}_k$. That is:

$$\bar{\lambda}_k = \frac{\sum_{i=1}^{P} \lambda_k^{(i)}}{P} \quad (29) \qquad\qquad \bar{d}_k = 10^{\bar{\lambda}_k}. \quad (30)$$

5.2. Scaling of the trajectory

This scale factor must be applied to the state variables with length dimensions, position and velocity. These are encapsulated in a vector:

$$\mathbf{x}_k^{C_d}(n) = \begin{bmatrix} \mathbf{r}_{W,k}^C(n) \\ \mathbf{v}_{W,k}^C(n) \end{bmatrix}. \tag{31}$$

To ensure the continuity in position and velocity, each vector $\mathbf{x}_k^{C_d}(n)$ contained in the current section k is scaled using the following recursive formula:

$$\breve{\mathbf{x}}_k^{C_d}(n) = \breve{\mathbf{x}}_{k-1}^{C_d}(N) + \bar{d}_k \left[\mathbf{x}_k^{C_d}(n) - \mathbf{x}_{k-1}^{C_d}(N)\right] \quad k = 2, 3, ... \tag{32}$$

$$\breve{\mathbf{x}}_1^{C_d}(n) = \bar{d}_1 \mathbf{x}_1^{C_d}(n), \tag{33}$$

where $\breve{\mathbf{x}}_k^{C_d}(n)$ contains the scaled camera position and velocity estimates.

5.3. Scaling of landmarks

The set of landmarks to be updated or initialised in the scaled map estimate is given by all the point landmarks marginalised from the EKF during current section and the landmarks matched in the EKF update of the last pose of the section:

$$\mathcal{M}_{S,k} = \left[\mathcal{E}^p_{(k-1)N+1}, \cdots, \mathcal{E}^p_{kN}, \mathcal{T}^p_{kN}\right]. \tag{34}$$

The scale of each landmark is set to the one of its anchor pose in IDP:

$$\breve{\mathbf{y}}_W^{(j)} = \breve{\mathbf{r}}_W^{C(j)} + \bar{d}_{\kappa(j)} \left(\mathbf{y}_W^{(j)} - \mathbf{r}_W^{C(j)}\right) \tag{35}$$

where the function $\kappa : \mathbb{N} \rightarrow \mathbb{N}$ which establishes a surjective mapping from a landmark index j to the index of the section containing the anchor pose $\mathbf{r}_W^{C(j)}$ of the landmark.

5.4. Real time implementation and reduction of the update delay

The complete scaling algorithm is implemented in a new thread within the real-time monoSLAM C++ application [50], working in parallel with the main SLAM thread. After each EKF iteration, the main thread stores the last camera pose and the corresponding landmarks in two shared buffers. When the buffer with the camera poses is fully updated with N poses, the main thread triggers the scaling thread. After executing the scaling algorithm, the scaled trajectory is updated by adding the recently scaled camera poses.

In spite of the real-time operation, the update of the scaled odometry estimate is delayed due to the time t_{DFT} it takes to fill the buffer of camera poses with the N states needed to perform the DFT. One way to reduce this delay is to reduce N, at the expense of reducing the accuracy of the DFT to compute the step frequency.

Alternatively we propose to use a sliding window, updating only one fraction N_f of the buffer of camera poses at a time. Thus the number of camera poses used for the spectral analysis remains N by reusing poses from previous sections, while the amount of scaled camera poses per section is reduced to N_f. The time required to update the scaled trajectory with the new N_f poses is notated as t_{upd}. The complete scaling method is described in Algorithm 1.

Algorithm 1 Complete Visual Odometry Scaling algorithm

```
Require: $\mathbf{x}_k^C(1..N)$, $\mathcal{S}_{k-1}$
Ensure: $\breve{\mathbf{x}}_k^C(1..N_f)$, $\mathcal{S}_k$
  //Notation
  $\mathbf{x}_k^C(n) =$ $n^{th}$ unscaled camera state of section $k$
  $\breve{\mathbf{x}}_k^C(n) =$ $n^{th}$ scaled camera state of section $k$
  $N =$ # input camera states
  $N_f =$ # output/new camera states
  $\mathcal{S}_k =$ Set of particles for the particle filter
  //Algorithm
  $k = 0$; $[\mathcal{S}_0]$ = Initialise particles ()
  while Not end of sequence do
    $k = k + 1$
    Wait for new $\breve{\mathbf{x}}_k^C(1..N)$ from monoSLAM
    $[z_k(1..N),\ \mu_{V,k},\ \sigma_{V,k}]$ = Extract $z$-comp & mean speed $\left(\mathbf{x}_k^C(1..N)\right)$
    $[z_k(1..N)]$ = High Pass Filter $(z_k(1..N))$
    $[f_m,\ \Gamma_{d,k}]$ = Spectrogram $(z_k(1..N))$
    $[f_{st,k},\ \Gamma_{d,k}(f_{st,k})]$ = Estimate Step Frequency $(f_m,\ \Gamma_{d,k})$
    if Step freq power is consistent $(d_{k-1},\ \Gamma_{d,k}(f_{st,k}))$ then
      $[V_{walk,k}]$ = Walking speed model $(f_{st,k})$
      $[\mathcal{S}_k]$ = Sample Proposal Distribution $(\mathcal{S}_{k-1},\ \mu_{V,k}, \sigma_{V,k})$
      $[\mathcal{S}_k]$ = Weighting and Resampling $(\mathcal{S}_k,\ V_{walk,k})$
      $[d_k]$ = Compute mean scale factor $(\mathcal{S}_k)$
    else
      $d_k = d_{k-1}$ ; $\mathcal{S}_k = \mathcal{S}_{k-1}$
    end if
    if k=1 then
      $\left[\breve{\mathbf{x}}_1^C(1..N)\right]$ = Scale Section $\left(d_1,\ \mathbf{x}_1^C(1..N)\right)$
    else
      $\left[\breve{\mathbf{x}}_k^C(1..N_f)\right]$ = Scale Section $\left(d_k,\ \mathbf{x}_k^C((N-N_f+1)..N)\right)$
    end if
  end while
```

6. Experiments

For the validation of our proposal we use three different cameras in our experiments, each one with varying geometries, resolution and frame rates.

The first experiments have been carried out with a catadioptric omnidirectional camera VS-C14U-80-ST model which consists on a conic mirror and a Sentech UltraSmall STC-MC83USB camera with a resolution of 1024x768, at frame rate of 15 fps, which is mounted on a helmet to be carried by a human operator.

In the last experiments we used the wearable GoPro Hero and Sony Action Cam, in order to verify the applicability of our method also to conventional cameras, whose narrower field of view is likely to provide less accurate motion estimates than omnidirectional cameras [51], which could affect the perception of the body oscillations. Also, their ability to be attached to several body parts, allows us also to evaluate our method when the camera is worn on the chest, which, as in the case of the head, also shows the characteristic oscillatory motion while walking.

Trajectory estimates in the experiments have been refined by applying loop closure optimisation when possible, where the loop constraints were determined as explained in 3.2. Frames for loop closure were selected manually, since the loop detection problem is out of the scope of this work. Also this avoids uncertainties due to possible errors in automatic loop detection, allowing for a fair comparison of the scale drift removal capabilities between our method and the proposed in [8].

The Ground Truth for the experiments has been obtained from the Google Maps satellite view using the distance measurement tool. To compare numerically the Ground Truth and the scaled odometry estimates, we parametrise both curves by the normalised arc length or accumulated distance ξ which spans from 0 (start) to 1 (end). Then, given the Ground Truth trajectory $\mathbf{t}_{GT}$, the error for a given pose is computed as follows:

$$err^{(i)} = \left\|\mathbf{t}_{VO}^{(i)} - \mathbf{t}_{GT}(\xi(\mathbf{t}_{VO}^{(i)}))\right\|. \tag{36}$$

6.1. Parameter tuning

Our algorithm presents a series of parameters that have to be tuned previously. The number of particles used in the particle filter is set to $P = 5000$ for all the experiments. The standard deviation of the distribution for the initial logarithmic scale factor is set to $\sigma_0 = 1$, which allows us to cover an uncertainty interval for the scale factor between 10^{-2} and 10^{2} with a 95% of confidence. The setup of the uncertainty parameters for the scale drift σ_{drift} and $\sigma_{V_{walk}}$ is heuristic, trying to reach a trade-off between robustness and rapid response to sharp scale factor changes. We use $\sigma_{drift} = 0.1$ and $\sigma_{V_{walk}} = 0.2$ m/s. This last value lies within the interval of standard deviations for the feasible step frequencies range, as it can be proved by computing the derivative of (11):

$$\sigma_{V_{walk}} = \alpha\beta f^{\beta-1} H \sigma_f, \tag{37}$$

with $\sigma_f = \frac{1}{t_{DFT}}$; and taking $t_{DFT} = 3$ s and evaluating at the limits of the feasible step frequencies. Note that though for larger t_{DFT}, $\sigma_{V_{walk}}$ should be theoretically lower, keeping it constant overcompensates the fact of loosing accuracy due to a lower robustness to changes in speed when using larger windows.

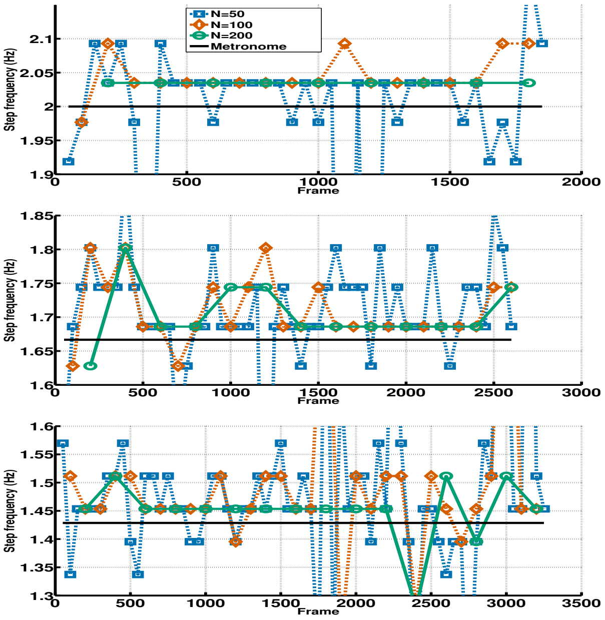


Figure 6. Spectral analysis along the same path at the three step frequencies of 2 (top), 1.67 (centre) and 1.43 Hz (bottom) with different section sizes. A higher section size implies a more reliable frequency estimate.

6.2. Testing the ability to measure the step frequency

For the first experiment we acquired three image sequences with the catadioptric camera walking along the same path of 230 meters with three different step frequencies. In the same way as in the process detailed in Section 4.1 the user's pace was synchronised with the beep of a metronome with 0.01 seconds of resolution. The metronome was set up to 0.50, 0.60 and 0.70 seconds per beat for each sequence, which translates in step frequencies of 2, 1.67 and 1.43 Hz respectively. The purpose of this experiment is testing the ability of the method described in Section 4.2 to estimate the step frequency just from the visual odometry signal.

We select different section dimensions of $N = 50$, $N = 100$ and $N = 200$ camera poses. To compute the DFT we use the FFTW (Fast Fourier Transform West) C library [52].

In Fig. 6 it is shown that the dominant frequency obtained by spectral analysis closely approximates the step frequency fixed by the metronome. Among the considered setups, $N = 200$ results in better accuracy and less outliers in the estimation of the step frequency. However, as argued in Section 4.2, a window of this length might not be the optimal for the performance of our method as a whole. This issue is addressed in the next experiments.

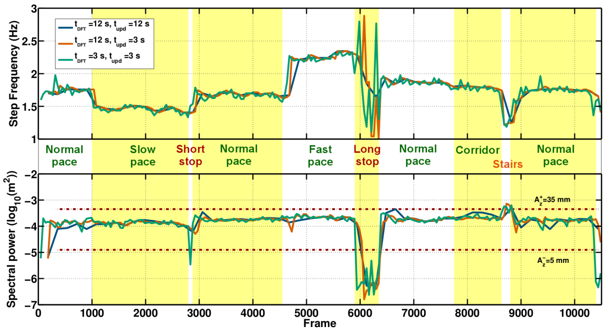


Figure 7. Evolution of the step frequency estimate (top) and its corresponding spectral power (bottom) in the changing pace experiment. Consistency bounds are violated when estimate does not correspond to a walking step frequency.

6.3. System robustness under changes of pace

The sequence for this experiment was acquired with the catadioptric camera along a path of 886 m containing a variety of challenging situations like changes of pace, stops, stairs and walking along a narrow corridor. Transitions between these situations have been ticked accordingly in the frame when they take place.

The objective is evaluating the robustness of our method under these conditions, and select the optimal number of poses to be taken for scaling at each iteration. Instead of using the number of poses N and N_f to define the window sizes in the experiments, from this point we will take their temporal length t_{DFT} and t_{upd}, as it generalises better for different frame rates.

In this experiment we evaluate the use of a dynamic window approach to scale the trajectory sections. We have tested 3 different alternatives. In two alternatives t_{DFT} is set to 12 seconds and t_{upd} is varied to compare the performance of static (t_{upd} = 12 seconds) and dynamic (t_{upd} = 3 seconds) windows. The third alternative consists in using a static window with $t_{DFT} = t_{upd} = 3$ seconds.

Fig. 7 (top) reveals how the step frequency computed from the raw unscaled visual odometry varies accordingly with the pace of the walker. Note that for lower t_{DFT} the step frequency estimate is less accurate and oscillates more, though the global tendency in the pace is still captured. In Fig. 7 (bottom), during the long stop, the spectral power shows a violation of the consistency condition which leads to ignore the erratic estimation of the step frequency. During the short stop, the violation of the consistency is not observed with t_{DFT} = 12 s, due to the masking effect of the poses corresponding to a walking state. In the case of going upstairs the power peak is noticeable but not as clear as in the stop to establish a clear upper limit for the consistency condition.

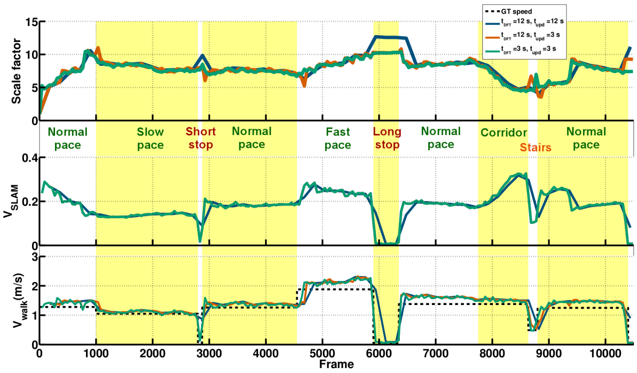


Figure 8. Evolution of (top) the scale factor, (middle) the non-dimensional speed and (bottom) the estimated real walking speed in the changing pace experiment. Ground Truth speed was assumed constant for a given pace and computed from Google Maps Ground Truth and the time difference between frames.

Table 1. Estimation error for combinations of t_{DFT} and t_{upd} for the experiment with changes in pace.

t_{DFT} [s]	t_{upd} [s]	Mean error[m]	Maximum error[m]	Relative mean error
12	12	12.54	28.83	1.42%
12	3	10.43	18.65	1.18%
3	3	9.26	21.24	1.05%

Table 2. Estimation error for different scaling and optimisation combinations for the experiment with changes in pace.

Method	Mean error[m]	Maximum error[m]	Relative mean error
unscaled	30.47	71.10	3.43%
optimSim3 [8]	11.29	30.02	1.27%
dynScale (ours)	5.35	19.34	0.60%
dynScale (ours) + optimSim3 [8]	4.04	9.68	0.46%

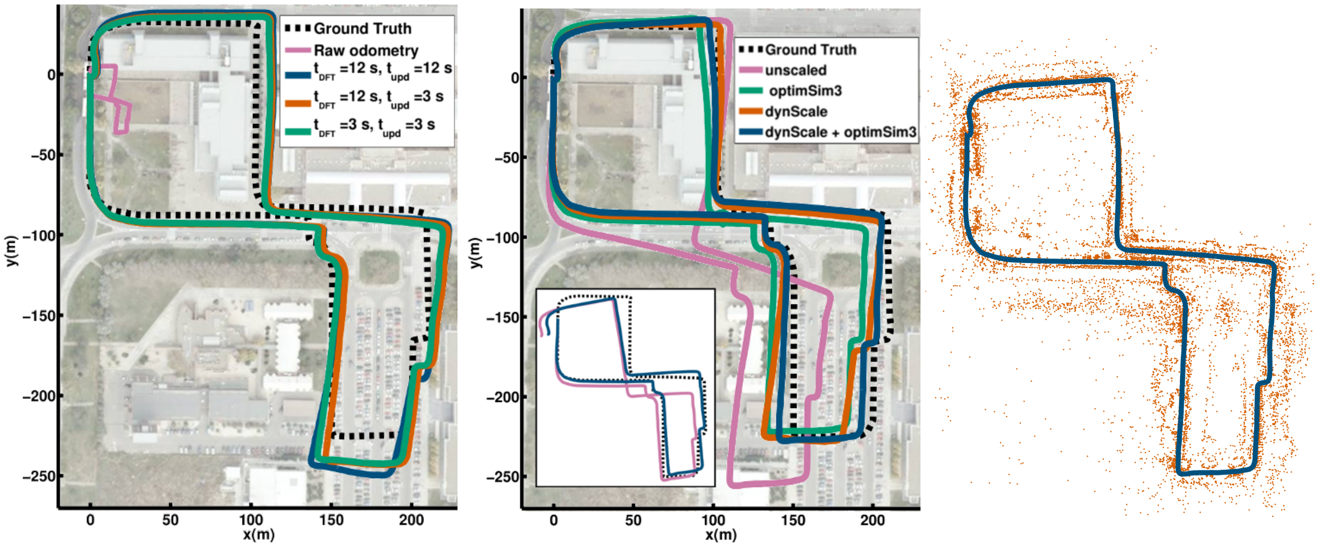
Fig. 8 (top) shows the dynamic estimation of the scale factor. It can be noticed that scale drift can occur in two different ways. On the one hand it can be a gradual drift as occurs at the start of the sequence or during the corridor. Drift at the start can be due to the still highly uncertain depth estimate of the tracked points during initialisation, while on the corridor it is the gradual substitution of points in spacious areas by points on the walls of the narrow corridor what causes the drift. On the other hand drift can occur sharply if landmarks which act as an anchor for the scale are suddenly lost. This might be caused for example by occlusions by dynamic elements or sudden camera accelerations, as occurs when restarting the walk after a stop.

Fig. 9a shows a comparison between the scaled trajectories obtained for each considered setup. Loops have been closed at the end and at the middle of the path. Numerical comparison with the Ground Truth is detailed in Table 1. It is shown that taking t_{DFT} = 3 s offers an slightly better scaled estimate of the visual odometry with a mean relative error of 1.05% over the trajectory length. This demonstrates the convenience of losing a bit of accuracy in the DFT using short windows (but large enough to capture the body oscillations), in order to get a higher update rate of the scale and a faster detection of stops or changes in pace (Fig. 7).

The great improvement of all the considered cases with respect to the raw odometry estimate shows the ability of our approach not only to compute the scale but also to remove the scale drift, which is reflected on the deformation of the raw estimate.

6.3.1. Comparison with state-of-the-art scaling approach

Once we pick the configuration with $t_{DFT} = t_{upd} = 3$ s we compare our method with the approach described in [8]. This



(a) (b) (c)

Figure 9. Changing pace experiment. (a)Scaled trajectory estimates for different setups of t_{DFT} and t_{upd}. (b) Result of using different approaches for scale drift removal, with all trajectories rescaled to the Ground Truth's scale. Estimates of our scaled and raw trajectory estimates prior to loop closure are shown in the small view, in which the raw estimate has been rescaled for better visualisation. (c) Scaled trajectory and scene points obtained with the most accurate approach.

approach aims to remove the scale drift in the trajectory during loop closure optimisation by expressing the camera poses and both the odometry and loop closure constraints as similarity transforms (Sim(3)) instead of as conventional rigid body motions (SE(3)). Note that this approach tackles the scale drift problem but it does not compute the absolute scale of the odometric estimate. Thus, for a fair comparison, each of the finally obtained trajectories by the different evaluated methods is rescaled so that its total distance is the same as that of the Ground Truth. We evaluate 4 different solutions: standard loop closure optimisation in SE(3) of the raw estimate (unscaled), scale drift correction with loop closure optimisation in Sim(3) (optimSim3 [8]), standard loop closure optimisation in SE(3) of our scaled estimate (dynScale) and loop closure optimisation in Sim(3) of our scaled estimate (dynScale + optimSim3 [8]). Qualitative and numerical results shown in Fig. 9b and Table 2 respectively. It can be observed that both our method and [8] clearly improve the raw estimate, though our method provides better accuracy. Note also that the combination of both methods slightly improves our raw proposal providing the most accurate estimate.

This result is expected since while our approach estimates the scale every 3 seconds, in [8] it is only possible to observe it at loop closures. However it can be noted that, as both approaches obtain scale drift information from different sources, they can combine well providing a more accurate estimate together than both alone. A complete view of the best trajectory estimate and the points of the scene is shown in Fig. 9c.

6.4. Indoor experiment

In the second experiment we test our approach with the catadioptric camera in an indoor environment [53] along a path of 464 m under a freely chosen gait, keeping the tuning used in previous experiment. Our approach is able to correct a significant scale drift of 200% taking place at the start of the trajectory (Fig. 10b), providing a final odometry estimate (Fig. 11a) with a mean error of 4.69 m (1.01% over the trajectory length). Note in Fig. 10a the peak in the spectral power occurring at the two stair parts in the trajectory.

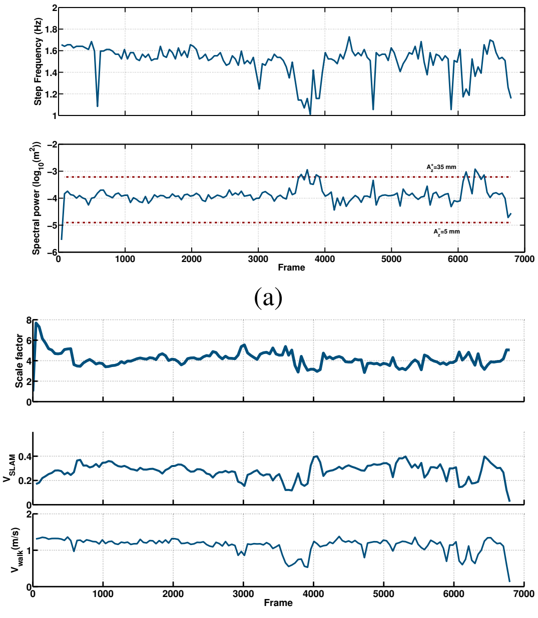
The comparison with [8] shown in Fig. 11 and Table 3 is similar to the previous experiment. Our approach alone outperforms [8] alone, and both together provide a better estimate.

6.5. GoPro experiment

For this experiment we acquired an image sequence with a GoPro camera attached to the user's head. The user walked at steady pace along a path of 410 m. the camera resolution and frame rate were set to 1280x720 and 60 fps respectively. The tuning of our scaling algorithm is the same as the one used in the previous experiments with the omnidirectional camera, taking again $t_{DFT} = t_{upd} = 3$ s.

In Fig.12a it is shown both the evolution of the step frequency and the spectral power of the head oscillation. Note that there exists a peak in the spectral power above the higher limit for oscillation amplitude, which corresponds to the stairs part of the trajectory. An outlier can be observed in the estimation of the step frequency, though, as it can be noted in Fig. 12b, the particle filter successfully rejects this measure and both scale factor and walking speed estimates are not affected. As in the previous experiment the scale smoothly drifts as we enter in the narrow corridor (between frames 2200 and 3200).

The final odometric estimate in Fig. 13a shows once again the competitiveness of our method, with a mean error of 1.79 m (0.44% over the total trajectory length). Note that even prior to loop closure, scale drift correction can be observed in the small view as the start and end points of the trajectory are almost coincident. In this experiment the loop was closed at the end of



(a)

(b)

Figure 10. Indoor experiment with catadioptric camera. (a) Evolution of the step frequency estimate (top) and its corresponding spectral power (bottom), (b) Evolution of (top) the scale factor, (middle) the non-dimensional speed and (bottom) the estimated real walking speed.

the sequence. As the camera view direction of the scene differs between the start and the end poses, and a camera with limited field of view is used, correspondences for loop closure can only be obtained in a small portion of the matched images, which result in a loss of precision of the loop constraint.

As it can be observed in Fig. 13b and Table 4, our approach outperforms alone clearly all the other options. The optimSim3 improves the performance of standard solution, but does not improve our scaling approach even when combined with our method, which may be due to the limited common field of view in the loop closing constraint. The final odometry and map view obtained with our approach is shown in Fig. 13c.

6.6. *Sony Action Cam attached to the chest*

In this experiment, we test the ability of our method for performing scale correction when the camera is not worn on the head. We used a Sony Action Cam with a resolution of 960x540 at 30 fps attached to the user's chest. The sequence was acquired in an indoor loop with a length of 187 m.

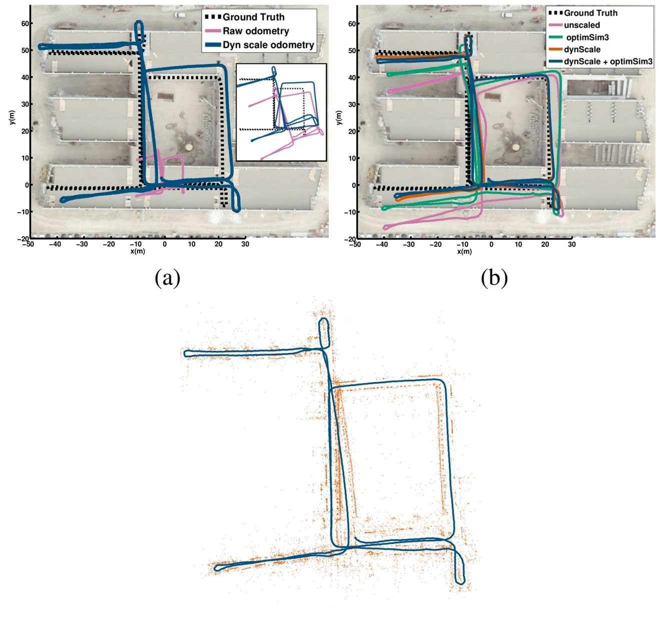
Fig. 14 show that the step frequency and the scale factor can be successfully estimated when the camera is worn on the chest, due to showing only a oscillatory motion during locomotion. The final reconstruction after loop closure (Fig. 15) shows the correction of a slight drift in the scale and that the estimate with the absolute scale is indeed computed.

6.7. *Analysis of the computational cost*

In Fig. 16 we show the computation time to perform the whole algorithm to scale each section of the visual odometry. After the initialization cost due to the memory allocation of variables required to compute the FFT, the computational cost stabilizes around 0.01 seconds. Given that the data required to compute the scale is delivered to the scaling thread with an update time t_{upd} not lower than 3 seconds as shown in previous experiments, our method fits with the real-time requeriments of a SLAM systems.

Table 3. Estimation error for different scaling and optimisation combinations for the indoor experiment with the catadioptric camera.

Method	Mean error[m]	Maximum error[m]	Relative mean error
unscaled	10.82	22.90	2.33%
optimSim3 [8]	6.47	12.82	1.39%
dynScale (ours)	3.77	10.37	0.81%
dynScale (ours) + optimSim3 [8]	3.21	9.37	0.69%



(a)

(b)

(c)

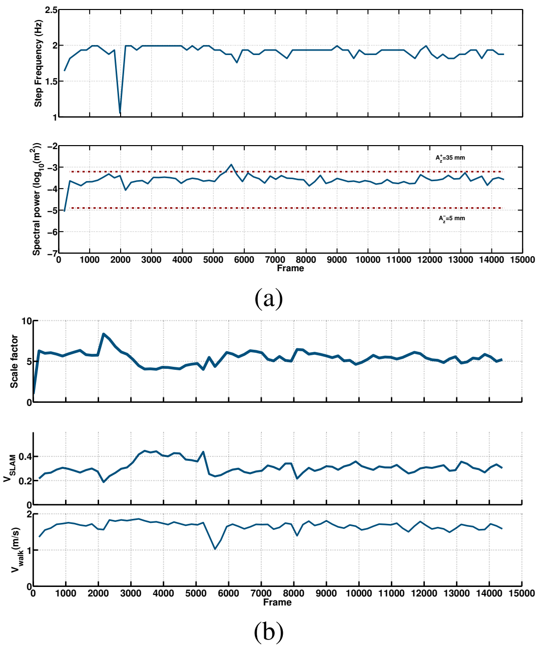
Figure 11. Indoor experiment with catadioptric camera. (a)Trajectory estimates after loop closure in our scaled estimate and the raw estimate. Estimates prior to loop closure are shown in the small view, in which the raw estimate has been rescaled for better visualisation. (b) Result of using different approaches for scale drift removal. Absolute scale of each estimate is fitted to the Ground Truth's. (c) Scaled trajectory and scene points obtained with our approach.

6.8. *Analysis of the change in the walking model parameters*

In this section we analyse how the scaled estimate of some of the previous experiments is affected by the variation of the walking model parameters α and β from their nominal values explicitly fitted for the user. We evaluate the effect both on the absolute scale and on the scale drift. A prior theoretical analysis is lead by naively assuming that:

$$d(t) = \frac{V_{walk}(t)}{V_{SLAM}(t)}. \tag{38}$$

The change of the model parameters produce a variation of the estimated V_{walk}, which leads to a change in d. The change in the absolute scale is expressed by the following ratio:



(a)

(b)

Figure 12. GoPro experiment. (a) Evolution of the step frequency estimate (top) and its corresponding spectral power (bottom), (b) Evolution of (top) the scale factor, (middle) the non-dimensional speed and (bottom) the estimated real walking speed.

$$r(t) = \frac{d(t)}{\hat{d}(t)} = \frac{V_{walk}(t)}{\hat{V}_{walk}(t)}, \tag{39}$$

while the scale drift with respect to the nominal model is given by its first order time derivative $r'(t)$.

Let us first analyse how the variation of parameter α affects both $r(t)$ and $r'(t)$. Recalling (11) we get that:

$$r(t) = \frac{\alpha}{\hat{\alpha}} \tag{40}$$

$$r'(t) = 0. \tag{41}$$

which means that a proportional change in the scale is to be expected if we vary α. Since α is a constant parameter, no drift is expected with respect to the nominal scaled estimate.

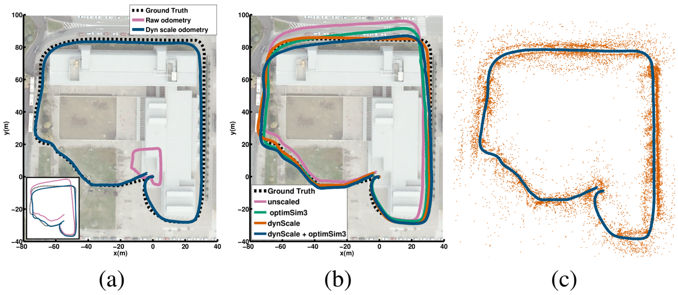
In the case of varying β we have:

$$r(t) = \frac{f^{\beta}}{\hat{f^{\beta}}} = f^{\Delta\beta} \tag{42}$$

$$r'(t) = \Delta\beta f^{\Delta\beta - 1}\frac{\partial f}{\partial t}, \tag{43}$$

which means first, that the change in the absolute scale will be higher with increasing $\Delta\beta$ and higher step frequencies and, secondly, that if there are changes in the user's pace, we might expect a drift in the scale higher for larger $\Delta\beta$.

For the experimental evaluation we used the changing pace and GoPro sequences. Each time we run our algorithm, we set one parameter to the nominal value fitted to our camera operator, while varying the other between our nominal value and 3 other options computed from average, lower limit and upper limit values of the parameters for the model proposed in [12].



(a)

(b)

(c)

Figure 13. GoPro experiment. (a)Trajectory estimates after loop closure in our scaled estimate and the raw estimate. Estimates prior to loop closure are shown in the small view, in which the raw estimate has been rescaled for better visualisation. (b) Result of using different approaches for scale drift removal. Absolute scale of each estimate is fitted to the Ground Truth's. (c) Scaled trajectory and scene points obtained with our approach.

Table 4. Estimation error for different scaling and optimisation combinations for the GoPro sequence.

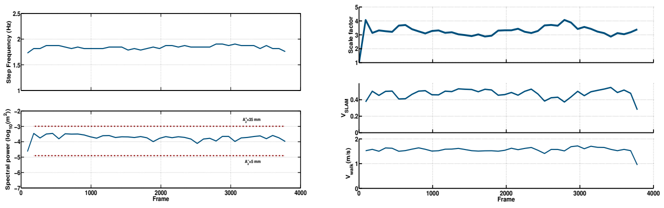
Method	Mean error[m]	Maximum error[m]	Relative mean error
unscaled	5.83	11.94	1.42%
optimSim3 [8]	2.81	7.40	0.69%
dynScale (ours)	1.54	4.46	0.37%
dynScale (ours) + optimSim3 [8]	2.53	5.89	0.62%

Results in Fig. 17 show that an *ad hoc* calibration for each user is crucial to get the absolute scale of the estimate. However, for just scale drift correction, which is more important in practice, it is shown that an accurate calibration of the walking model parameters is not critical, as long as extreme pace changes are avoided during walking. Note that the experimental observations confirm the predicted behaviour by the theoretical analysis.

7. Conclusions

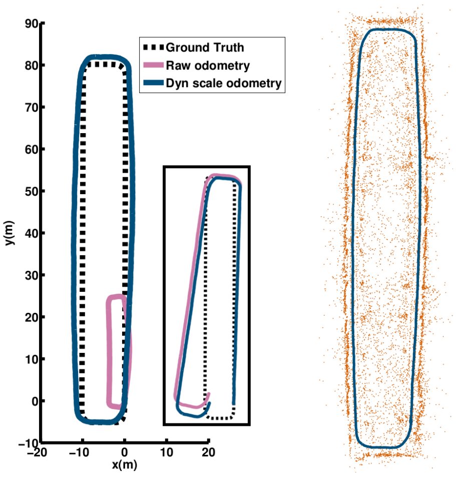
In this paper we have presented a novel approach to provide estimates with the absolute scale of wearable odometric localisation systems using a single camera. Our proposal makes these hypothesis: first, the camera must be attached to a body part whose motion is mainly caused by the action of walking; secondly, the initial unscaled visual odometry estimate must be accurate enough to register the oscillations which take place during walking, and thirdly the roughness of the terrain on which the user moves is low enough not to mask the amplitude of the walking oscillations.

Our method has been thoroughly evaluated in a rich set of video sequences, combining indoor and outdoor environments and using many kinds of cameras, attached either the head or to the chest of the user. In spite of this high variety in the conditions which the system has been tested on, our algorithm shows a good performance without requiring to be retuned for each experiment with the same user. Also we show that if the pace of the user does not change a lot during the path, the calibration of our system for a specific user is not critical in terms of scale drift correction.



(a) (b)

Figure 14. Camera-on-chest experiment. (a) Evolution of the step frequency estimate (top) and its corresponding spectral power (bottom), (b) Evolution of (top) the scale factor, (middle) the non-dimensional speed and (bottom) the estimated real walking speed.



(a) (b)

Figure 15. Camera-on-chest experiment. (a)Trajectory estimate after loop closure with and without our scaling algorithm. Estimates prior to loop closure are shown in the small view, in which the raw estimate has been rescaled for better visualisation. (b) Scaled trajectory and scene points obtained with our approach.

We have compared our algorithm in our most challenging sequences against the scale drift correction method proposed in [8], where ours shows a better performance. This gain in performance is due to the fact that our method corrects the scale dynamically every few seconds, instead of having to wait for a loop detection to obtain scale information as done in [8]. Note also, that as both methods extract the scale from different sources, they are not mutually exclusive. Indeed we have shown in the experiments, that he combination of both provides the better performance. In this sense, we expect also that the proposal presented in this paper can also combine well not only with [8], but also some of other present and future methods for scale computation, in order to get more robust information about the real camera trajectory and 3D observed scene.

8. Acknowledgments

This work was supported by Ministerio de Economía y Competitividad and European Union (project DPI2014-61792-EXP) and Ministerio de Educación, Cultura y Deporte (scholarship FPU AP2012-5507)

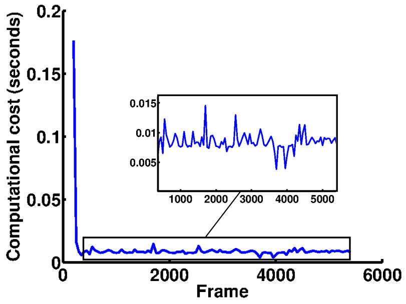


Figure 16. Computation time used to perform our approach for the different sections of the indoor sequence.

References

[1] S. Mann, Smart clothing: The wearable computer and wearcam, Personal Technologies 1 (1) (1997) 21–27.

[2] S. Hodges, E. Berry, K. Wood, Sensecam: a wearable camera that stimulates and rehabilitates autobiographical memory., Memory 19 (7) (2011) 685–96.

[3] S. Mann, J. Huang, R. Janzen, R. Lo, V. Rampersad, A. Chen, T. Doha, Blind navigation with a wearable range camera and vibrotactile helmet, in: ACM Int. Conf. on Multimedia (ICMM), 2011, pp. 1325–1328.

[4] A. J. Davison, Real-time simultaneous localisation and mapping with a single camera, in: International Conference on Computer Vision, Vol. 2, 2003, pp. 1403–1410.

[5] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, Fastslam: A factored solution to the simultaneous localization and mapping problem, in: In AAAI National Conference on Artificial Intelligence, 2002, pp. 593–598.

[6] G. Klein, D. Murray, Parallel tracking and mapping for small AR workspaces, in: IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR), 2007, pp. 225–234.

[7] C. Wu, Towards linear-time incremental structure from motion, in: 3DV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DV-CON), 2013, pp. 127–134.

[8] H. Strasdat, J. M. M. Montiel, A. Davison, Scale drift-aware large scale monocular slam, in: Robotics: Science and Systems (RSS), 2010.

[9] J. Civera, A. J. Davison, J. M. M. Montiel, Inverse depth parametrization for monocular slam, IEEE Trans. on Robotics 24 (5) (2008) 932–945.

[10] J. Civera, O. G. Grasa, A. J. Davison, J. M. M. Montiel, 1-Point RANSAC for EKF Filtering: application to real-time structure from motion and visual odometry, J. of Field Robotics 27 (5) (2010) 609–631.

[11] R. Mur-Artal, J. M. M. Montiel, J. D. Tardós, ORB-SLAM: A versatile and accurate monocular SLAM system, IEEE Trans. on Robotics (T-RO) 31 (5) (2015) 1147–1163.

[12] D. Grieve, R. J. Gear, The relationships between length of stride, step frequency, time of swing and speed of walking for children and adults, Ergonomics 5 (9) (1966) 379–399.

[13] M. Zarrugh, F. Todd, H. Ralston, Optimization of energy expenditure during level walking, European Journal of Applied Physiology and Occupational Physiology 33 (1974) 293–306.

[14] A. D. Kuo, A simple model of bipedal walking predicts the preferred speed-step length relationship, Journal of Biomechanical Engineering 123 (2001) 264–269.

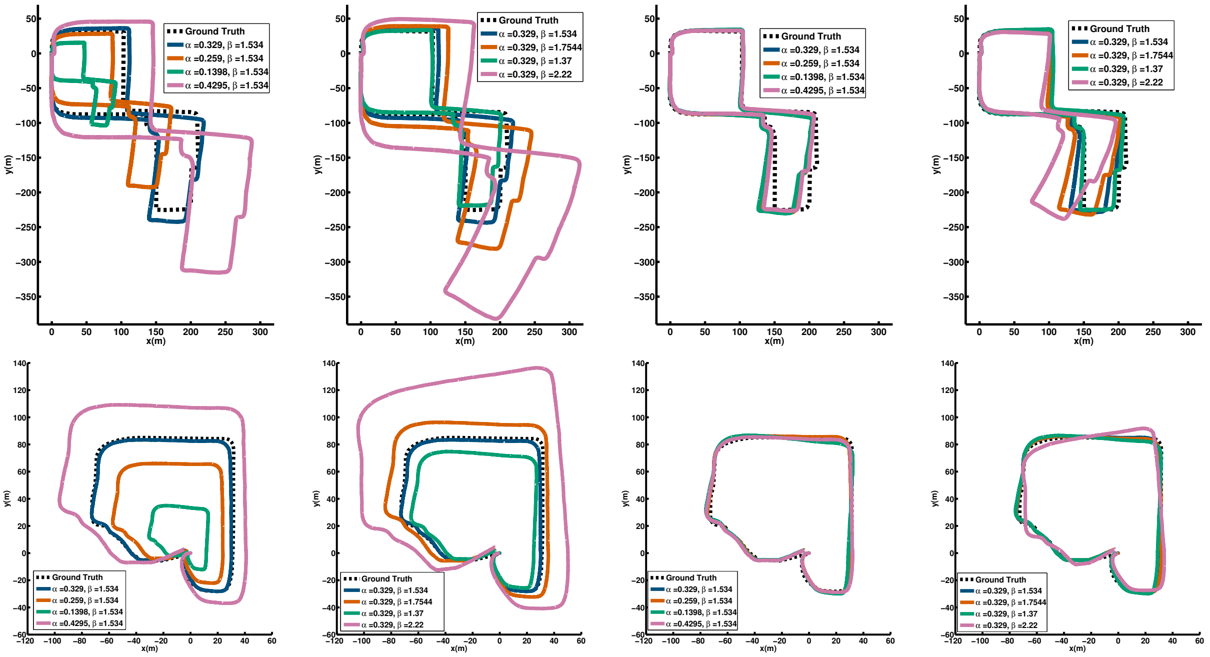
[15] D. Gutiérrez-Gómez, L. Puig, J. J. Guerrero, Full scaled 3d visual odometry from a single wearable omnidirectional camera, in: IEEE/RSJ Int. Conf. on Intelligent Robot Systems (IROS), 2012, pp. 4276–4281.

[16] D. Gutiérrez-Gómez, J. J. Guerrero, Scaled monocular slam for walking people, in: Int. Symp. on Wearable Computing (ISWC), 2013, pp. 9–12.

[17] W. W. Mayol-Cuevas, B. J. Tordoff, D. W. Murray, On the choice and placement of wearable vision sensors, IEEE Trans. on Systems Man and Cybernetics Part A 39 (2) (2009) 414–425.

[18] X. Ren, C. Gu, Figure-ground segmentation improves handled object recognition in egocentric video, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3137–3144.

[19] A. Fathi, X. Ren, J. M. Rehg, Learning to recognize objects in egocentric



(a) (b) (c) (d)

Figure 17. Variation of (a) the absolute scale with α, (b) the absolute scale with β, (c) the scale drift with α, (d) the scale drift with β. Results for the pace change sequence are shown in the first row. Results for the GoPro sequence are shown in the second row row. Note that a wrong α has no negative effect on scale drift correction, while a wrong β is only harmful if there are high changes in pace.

activities, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3281–3288.

[20] H. Pirsiavash, D. Ramanan, Detecting activities of daily living in first-person camera views, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2847–2854.

[21] O. Aghazadeh, J. Sullivan, S. Carlsson, Novelty detection from an egocentric perspective, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3297–3304.

[22] K. M. Kitani, T. Okabe, Y. Sato, A. Sugimoto, Fast unsupervised egoaction learning for first-person sports videos, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3241–3248.

[23] D. Mitzel, B. Leibe, Close-range human detection for head-mounted cameras, in: British Machine Vision Conf. (BMVC), 2012.

[24] A. Fathi, J. K. Hodgins, J. M. Rehg, Social interactions: A first-person perspective, in: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1226–1233.

[25] M. Kourogi, T. Kurata, K. Sakaue., A panorama-based method of personal positioning and orientation and its real-time applications for wearable computers, in: Int. Symp. on Wearable Computing (ISWC), 2001, pp. 107–114.

[26] H. Aoki, B. Schiele, A. Pentland, Realtime personal positioning system for wearable computers, in: Int. Symp. on Wearable Computing (ISWC), 1999, pp. 37–43.

[27] W. W. Mayol, A. J. Davison, B. J. Tordoff, D. W. Murray, Applying active vision and slam to wearables, in: In Int. Symp. on Robotics Research (ISRR), 2003, pp. 325–334.

[28] R. O. Castle, G. Klein, D. W. Murray, Combining monoslam with object recognition for scene augmentation using a wearable camera, Image and Vision Computing (IVC) 28 (11) (2010) 1548–1556.

[29] H. Badino, T. Kanade, A head-wearable short-baseline stereo system for the simultaneous estimation of structure and motion, in: IAPR Conference on Machine Vision Applications (MVA), 2011.

[30] P. F. Alcantarilla, J. J. Yebes, J. Almazán, L. M. Bergasa, On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments, in: IEEE Int. Conf. on Robotics and Automation (ICRA), 2012, pp. 1290–1297.

[31] J. A. Hesch, S. I. Roumeliotis, Design and analysis of a portable indoor localization aid for the visually impaired, Int. J. of Robotics Research (IJRR) 29 (11) (2010) 1400–1415.

[32] M. Baglietto, A. Sgorbissa, D. Verda, R. Zaccaria, Human navigation and mapping with a 6 dof imu and a laser scanner, Robotics and Autonomous Systems (RAS) 59 (12) (2011) 1060–1069.

[33] T. Lupton, S. Sukkarieh, Removing scale biases and ambiguity from 6dof monocular slam using inertial., in: Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), 2008, pp. 3698–3703.

[34] G. Nützi, S. Weiss, D. Scaramuzza, R. Siegwart, Fusion of imu and vision for absolute scale estimation in monocular slam, J. of Intelligent Robotic Systems 61 (1-4) (2010) 287–299.

[35] J. Engel, J. Sturm, D. Cremers, Camera-based navigation of a low-cost quadrocopter, in: IEEE/RSJ Int. Conf. on Intelligent Robot Systems (IROS), 2012, pp. 2815–2821.

[36] S. Cumani, A. Denasi, A. Guiducci, G. Quaglia, Integrating monocular vision and odometry for slam., WSEAS Trans. on Computers 3 (2004) 625–630.

[37] A. Eudes, M. Lhuillier, S. Naudet-Collette, M. Dhome, Fast odometry integration in local bundle adjustment-based visual slam, in: International Conference on Pattern Recognition (ICPR), 2010, pp. 290–293.

[38] D. Scaramuzza, F. Fraundorfer, R. Siegwart, Real-time monocular visual odometry for on- road vehicles with 1-point ransac, in: Int. Conf. on Robotics and Automation (ICRA), 2009, pp. 4293–4299.

[39] P. Lothe, S. Bourgeois, E. Royer, M. Dhome, S. Naudet-Collette, Real-time vehicle global localisation with a single camera in dense urban areas: Exploitation of coarse 3d city models., in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 863–870.

[40] S. Song, M. Chandraker, Robust scale estimation in real-time monocular sfm for autonomous driving, 2014.

[41] D. Scaramuzza, R. Siegwart, Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles, IEEE Trans. on Robotics (T-RO) 24 (5) (2008) 1015–1026.

[42] D. Scaramuzza, F. Fraundorfer, M. Pollefeys, R. Siegwart, Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints, in: IEEE Int. Conf. on Computer Vision (ICCV), 2009, pp. 1413–1419.

[43] T. Botterill, S. Mills, R. Green, Correcting scale drift by object recognition in single camera slam, IEEE Trans. on Systems, Man, and Cybernetics–Part B: Cybernetics (2012) 1767–1780.

[44] P. Hansen, H. Alismail, P. Rander, B. Browning, Monocular visual odometry for robot localization in lng pipes, in: IEEE Int. Conf. on Robotics and Automation (ICRA), 2011, pp. 3111–3116.

[45] D. Gutierrez, A. Rituerto, J. M. M. Montiel, J. J. Guerrero, Adapting a real-time monocular visual slam from conventional to omnidirectional cameras, in: IEEE Int. Conf. on Computer Vision Workshops (ICCV), 2011.

[46] C. Geyer, K. Daniilidis, A unifying theory for central panoramic systems and practical applications, in: European Conf. on Computer Vision (ECCV), 2000, pp. 445–461.

[47] G. Bradski, The opencv library, Dr. Dobb's Journal of Software Tools 25 (11).

[48] L. Kneip, P. Furgale, OpenGV: A unified and generalized approach to real-time calibrated geometric vision, in: IEEE Int. Conf. on Robotics and Automation (ICRA), 2014.

[49] N. J. Gordon, D. J. Salmond, A. F. M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, Radar and Signal Processing, IEE Proceedings F 140 (2) (1993) 107–113.

[50] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, Monoslam: Real-time single camera slam, IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) 29 (2007) 1052–1067.

[51] A. Rituerto, L. Puig, J. J. Guerrero, Visual slam with an omnidirectional camera, in: Int. Conf. on Pattern Recognition (ICPR), 348–351, 2010, pp. 348–351.

[52] M. Frigo, S. G. Johnson, The design and implementation of FFTW3, Proceedings of the IEEE 93 (2) (2005) 216–231, special issue on "Program Generation, Optimization, and Platform Adaptation".

[53] A. C. Murillo, D. Gutiérrez-Gómez, A. Rituerto, L. Puig, J. J. Guerrero, Wearable omnidirectional vision system for personal localization and guidance, in: 2nd IEEE Workshop on Egocentric (First-Person) Vision, held with CVPR, 2012.