

Computing Egomotion with Local Loop Closures for Egocentric Videos

Suvam Patra *
IIT Delhi

Himani Arora * †
Columbia University

Himanshu Aggarwal * †
Qualcomm, Hyderabad

Subhashis Banerjee
IIT Delhi

Chetan Arora
IIIT Delhi

Abstract

Finding the camera pose is an important step in many egocentric video applications. It has been widely reported that, state of the art SLAM algorithms fail on egocentric videos [1, 2, 3, 4]. In this paper, we propose a robust method for camera pose estimation, designed specifically for egocentric videos. In an egocentric video, the camera views the same scene point multiple times as the wearer’s head sweeps back and forth. We use this specific motion profile to perform short loop closures aligned with wearer’s footsteps. For egocentric videos, depth estimation is usually noisy. In an important departure, we use 2D computations for rotation averaging which do not rely upon depth estimates. The two modification results in much more stable algorithm as is evident from our experiments on various egocentric video datasets for different egocentric applications. The proposed algorithm resolves a long standing problem in egocentric vision and unlocks new usage scenarios for future applications.

1. Introduction

Ability to record hands-free videos led to initial popularity of wearable cameras like Google glass [6] and GoPro [7]. These wearable cameras are typically harnessed to the wearer’s head, and provide a first person (wearer’s) view of the surroundings. The captured videos are therefore also referred to as egocentric videos. With the improvement in design and usability of egocentric cameras, such cameras are now being used for life logging, law enforcement, assistive vision as well as smart home applications.

Owing to their unique first person perspective and always-on nature, egocentric videos are rich in content usually unseen from point and shoot videos. This has generated lots of interest among the computer vision community. Computer vision researchers are mining egocentric videos

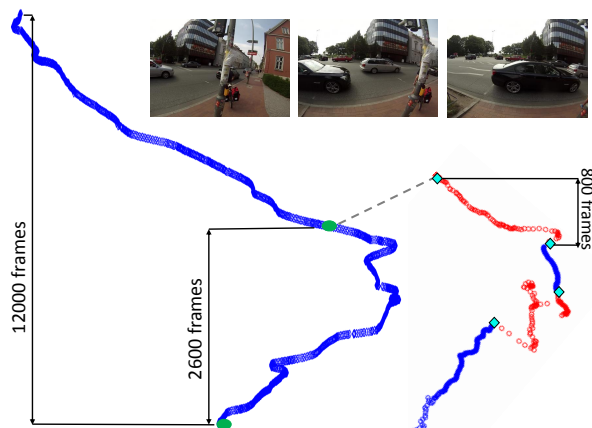


Figure 1. We present an egomotion estimation technique targeted for egocentric videos. The figure on the right shows a trajectory computed from LSD-SLAM [5] on a sequence from [2]. The different color segments show breaks in the trajectory computation every few 100 frames. On the other hand our algorithm runs successfully even after 12000 frames (on the left). The proposed method resolves a long standing problem in egocentric vision.

for understanding wearer’s activities [8, 9, 10, 11, 12], gaze fixation [3], temporal segmentation [3], video summarization [13, 14, 15] etc. In many of these applications, computing the pose of the camera and hence the position and gaze of the wearer is important.

For a long time, the egocentric community has struggled with the problem of egomotion estimation [1, 2, 3, 4]. Though visual odometry algorithms [5, 16, 17, 18, 19] have been quite successful for handheld or vehicle mounted cameras, the relatively ‘wild’ nature of egocentric videos: unrestrained camera motion, wide variety of indoor and outdoor scenes as well as moving objects, makes it extremely challenging for the egomotion estimation algorithms. Head rotation of the wearer causes the scene to change rapidly making any kind of feature tracking in egocentric video a non-trivial task. Any error in tracking causes the feature based visual odometry methods to lose putative matches,

*Equal Contribution

†Research assistant at IIIT Delhi during this project

and ultimately lead to their failure. Even when the tracking is not lost, sharp rotations introduce inaccuracies in pose estimates, and, in the absence of any recovery mechanism, the estimations drift to points of instability creating breaks in the trajectory. The typical motion profile of linear forward velocity due to wearer walking/driving in egocentric videos, kills any standard loop closure opportunity relying on similar camera pose.

An alternate approach is to employ the more robust dense or semi-dense SLAM¹ [5, 16, 17] techniques which utilize a greater portion of the image. Even though these methods perform better than their sparse counterparts, they are still unable to handle the wild dynamics of egocentric videos.

It is to be noted that shaky handheld videos **do not have motion profiles similar to** egocentric videos. For example, even a fast moving hand-held camera typically do not have dominant 3D rotation and usually scans a scene point multiple times giving standard loop closure opportunities.

In this paper we present a technique for camera egomotion estimation targeted especially for egocentric videos. We use these to correct pose inaccuracies using rotation averaging [20] and Gauss-Newton re-initialization. The specific contributions of the paper are as follows:

1. We present a fast and robust egomotion estimation method for egocentric videos.
2. We suggest a novel local loop closure technique aligned with the wearer’s head motion for egocentric videos. We exploit the typical to-and-fro sweeps in egocentric camera motion to detect local loop closures. While in all previous work for handheld videos, loop closures are typically carried out at the end, the proposed local loop closures aligned with the head sweep of the wearer enable fixing of estimation errors locally and efficiently. This not only makes the algorithm stable but also improves the accuracy significantly. The suggested improvement is specific to egocentric videos only.
3. While standard loop closure techniques typically fix both rotation and translation, the methodology fails in the presence of noisy depth estimates common in egocentric setting. We suggest to fix only rotation using 2D techniques which is both faster and more stable. We show the improvement in computed trajectory by fixing rotation alone.
4. We experiment on multiple problems such as EgoSampling [1], Hyperlapse [2], Temporal Segmentation [3], Activity Recognition [4] and Gaze Fixation [3]. For

each such problem the state-of-the-art egomotion estimation has been reported to fail [1, 2, 3, 4]. However the proposed method works flawlessly on all of these, indicating the robustness and efficacy of the technique.

Fig. 3 gives a schematic representation of the proposed approach.

2. Related work

Visual odometry algorithms can be classified into two broad categories: stereo and monocular. Stereo based visual odometry algorithms can usually give information about the absolute scale of the world. However, when the distance to the scene is much larger compared to the stereo baseline, these methods give no real advantage and hence, monocular visual odometry algorithms are typically used. Monocular methods employ only a single camera and suffer from the problem of scale ambiguity. The scale may be resolved by the integration of sensors such as IMU or with scene calibration objects.

Further classification of these algorithms can be done on the basis of the techniques used for pose estimation, namely feature based methods, **dense** methods and hybrid methods that employ a combination of both. Feature-based methods mainly consist of two steps [18, 21]. In the first step, important features are extracted from the images and matched. In the second step, the camera poses are estimated using only these sparse feature points. The reduction in the number of points used in the estimation process greatly reduces the computational load thereby increasing the speed. However, a large part of useful information in the scene is ignored.

Dense methods use the entire image and not just few selected features [19]. The camera poses are estimated as the set of parameters which minimize the image difference over all pixels in the image. To increase the accuracy of estimation, semi-dense methods are usually adopted, which perform photometric error minimization only in regions of sufficient gradient [5, 17].

The work closest to ours is LSD-SLAM [5] which does dense tracking directly on $sim(3)$, to explicitly detect scale-drifts. LSD-SLAM builds upon [17], to continuously estimate a semi-dense inverse depth map for the current frame, which in turn is used to track the motion of the camera using dense image alignment. Given an inverse depth map, both methods estimate the camera motion using non-linear minimization in combination with a coarse-to-fine scheme, as originally proposed in [16]. The minimization is done using weighted Gauss-Newton optimization on Lie-Manifolds. Our method also uses a similar optimization technique for the initial camera pose estimation.

As discussed in detail in [22], loop closures are detected using three major approaches in literature - map-to-map, image-to-image and image-to-map. [23] uses a map-to-map

¹Because of their overlap, we will be using SLAM/Odometry techniques interchangeably in our discussions, though our focus is only on the egomotion estimation and not on the associated 3D map.

approach where they find correspondences between common features in two sub-maps. [24] uses visual features for image-to-image loop-closures. Matching is performed based on presence or absence of these features from a visual vocabulary. [25] uses an image-to-map approach and finds loop-closure using re-localization of camera by estimating the pose relative to map correspondences. **Using the assumption that aerial video views a roughly planar ground and homographies can be used to register the frames, Leotta et al. [26] has proposed a homography-guided loop closure algorithm to address the long-term loop closure problem.** In this paper, we use an image-to-image approach for the purpose of detecting loop closures.

3. Background

The 6 DOF camera pose of an image I' w.r.t a reference image I refers to the rotation and translation of its camera centre. The camera pose is represented as an element of a Lie Group by a 4×4 matrix $\mathbf{g} \in \text{SE}(3)$ which comprises of a 3×3 rotation matrix $\mathbf{R} \in \text{SO}(3)$ and a 3×1 translation vector $\mathbf{t}$: $\mathbf{g} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$. In Lie Algebra, the camera pose is represented by a 6 element vector $\boldsymbol{\xi} \in \mathfrak{se}(3)$: $\boldsymbol{\xi} = [\mathbf{w} \quad \mathbf{v}]$, where, $\mathbf{w}$ and $\mathbf{v}$ represent the rotation and translation respectively.

We estimate the camera pose by aligning an image I' with a reference image I which minimizes the photometric error between the two image frames. Following the notation given in [16], the problem can be reduced to one of minimizing the sum $\sum_{i \in I} r_i$, where, the residual r_i , at a pixel i is given as: $r_i = I'(\tau(\boldsymbol{\xi}, \mathbf{x}_i)) - I(\mathbf{x}_i)$. Here $\boldsymbol{\xi}$ is the estimated camera pose of I' w.r.t I and $\tau : \mathbf{x} \rightarrow \mathbf{x}'$ is the warping function mapping a pixel in I to a pixel in I' .

The warping function can be constructed as follows. Using the inverse projection function, π , the 3D world point corresponding to a pixel $\mathbf{x}$ is given as: $\mathbf{X} = \pi^{-1}(\mathbf{x}, Z(\mathbf{x}))$, where $Z(x)$ is the depth at a pixel $\mathbf{x}$. When $\mathbf{X}$ is viewed in the coordinate frame of the second camera, it is transformed by the pose $\mathbf{g} = \exp(\boldsymbol{\xi})$ of I' w.r.t I to get the 3D world point $\mathbf{X}'$ given by $\mathbf{X}' = T(\boldsymbol{\xi}, \mathbf{X}) = \mathbf{R}\mathbf{X} + \mathbf{t}$. When $\mathbf{X}'$ is projected on the second camera, we get the warped pixel coordinates $\mathbf{x}' = \pi(\mathbf{X}')$.

3.1. Pose Estimation

The camera pose $\boldsymbol{\xi}$ can be computed as the one minimizing the residual: $\arg \min_{\boldsymbol{\xi}} \sum_{x_i} w_i(r_i(\boldsymbol{\xi}))^2$. The residual is calculated over all pixels x_i having valid depth estimates. The weight w_i at a pixel indicates the confidence measure of the estimated depth. The objective function is non-linear (since τ is non-linear in $\boldsymbol{\xi}$) and can be solved using Gauss-Newton optimization. We assume that the current estimate of $\boldsymbol{\xi}$ is known and iteratively refine the esti-

mate as $\boldsymbol{\xi} \leftarrow \boldsymbol{\xi} + \Delta\boldsymbol{\xi}$, where $\Delta\boldsymbol{\xi}$ is computed as:

$$\arg \min_{\Delta\boldsymbol{\xi}} \sum_{\mathbf{x}} w_i [I'(\tau(\mathbf{x}; \boldsymbol{\xi} + \Delta\boldsymbol{\xi})) - I(\mathbf{x})]^2 = \sum_{\mathbf{x}} w_i \underbrace{[I'(\tau(\mathbf{x}; \boldsymbol{\xi})) + \nabla I' \frac{\partial \tau}{\partial \boldsymbol{\xi}} \Delta\boldsymbol{\xi} - I(\mathbf{x})]^2}_{\text{First order Taylor approximation}} \quad (1)$$

where, $\nabla I'$ is the image gradient $\begin{bmatrix} \frac{\partial I'}{\partial x} & \frac{\partial I'}{\partial y} \end{bmatrix}$ and $\frac{\partial \tau}{\partial \boldsymbol{\xi}}$ is the Jacobian of the warping function.

The pose of each image frame I' is calculated with respect to a keyframe I . The poses thus obtained are concatenated to estimate the pose with respect to the first keyframe (assumed to be the world origin with $\boldsymbol{\xi} = 0$). Since only the depth corresponding to the reference frame I is required in estimating the warping function, only a keyframe is associated with a depth map. The keyframes are switched at regular intervals to maintain sufficient overlap between the frames.

3.2. Depth Map Estimation

The previous section assumed that the depth at a pixel $Z(\mathbf{x})$ is known apriori. We jointly estimate both the camera pose and the scene depth by computing stereo matches along epipolar lines at different values of depth. The SSD error is calculated at each of these pixels and the one with minimum error is chosen to be the correct match. Since initially both the depth and camera pose is unknown, the algorithm is bootstrapped with a random depth map having high weight only at those pixels where sufficient gradient is available. After a few keyframe propagation, the depth map converges to a reasonable approximation.

3.3. Rotation Averaging

Whenever the gaze direction returns to a neutral position following a deviation to the left or to the right, we detect local loop closures using a procedure outlined later in Section 4.1. The loop closures provide multiple redundant rotation estimates from the current frame to past frames - both through pair-wise estimates, and direct rotation estimates with a few archived key frames. The multiple redundant rotation estimates along different paths allow us to average out the rotations to obtain consistent rotations along all sub-paths. We denote by $\mathcal{C}$ the set of cameras in a window selected after loop-closure detection.

Let us assume that the relative rotation between camera i and camera j be denoted by $\mathbf{R}_{ij}$ and the absolute rotations be denoted by $\{\mathbf{R}_1, \dots, \mathbf{R}_N\}$ (where N is the number of cameras in the current window). Then the following relationship should hold:

$$\mathbf{R}_j \mathbf{R}_i^{-1} = \mathbf{R}_{ij} \quad (2)$$

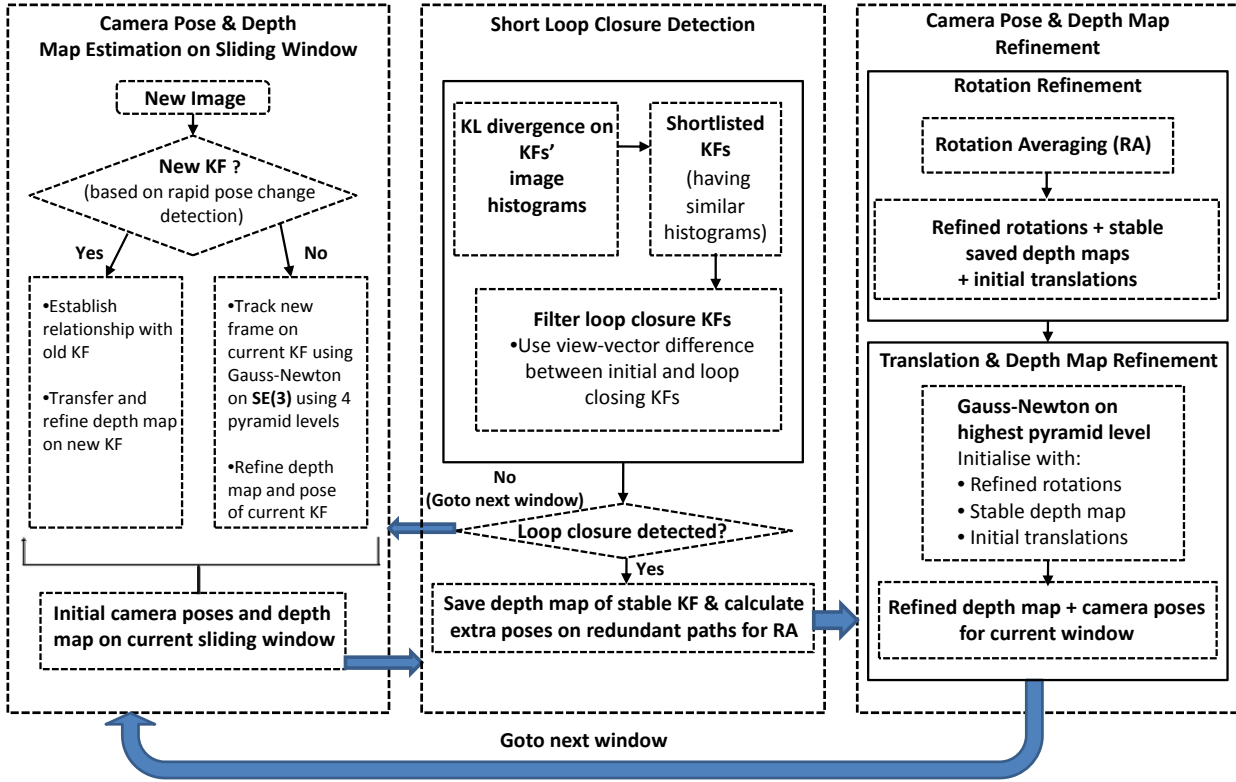


Figure 2. Our framework. (KF here denotes keyframes)

We estimate the absolute rotations $\{\mathbf{R}_1, \dots, \mathbf{R}_N\}$ by minimizing the sum of errors between the estimated pairwise rotations $\mathbf{R}_{ij}$ and the relative rotations calculated as $\mathbf{R}_j \mathbf{R}_i^{-1}$ from equation 2 as:

$$\{\mathbf{R}_1, \dots, \mathbf{R}_N\} = \arg \min_{\{\mathbf{R}_1, \dots, \mathbf{R}_N\}} \sum_{(i,j) \in \mathcal{E}} d^2(\mathbf{R}_j \mathbf{R}_i^{-1}, \mathbf{R}_{ij}) \quad (3)$$

where

$$d(\mathbf{R}_1, \mathbf{R}_2) = \frac{1}{\sqrt{2}} \|\log(\mathbf{R}_2 \mathbf{R}_1^{-1})\|_F \quad (4)$$

which is the intrinsic bi-variate distance measure defined on the manifold of 3D rotations or the Special Orthogonal group $SO(3)$ and $\|\cdot\|_F$ is the Frobenius norm. We use the robust function and the technique described in [20] to solve the optimization problem defined in Equation 3.

4. Proposed Algorithm

Our overall framework is depicted in Figure 2. In what follows we describe some of the key steps.

4.1. Short local loop closures

For a typical hand held video, global loop closure is an important step to fix the accumulated errors over individual

pose estimation. In egocentric setting, where the motion of the wearer is linear forward, a user may not revisit a particular scene point, which makes global loop closure impossible. Furthermore, given the usual noisy measurements due to wild nature of egocentric videos, the computed trajectories tend to drift quickly unless fixed by loop closures. We note that, a wearer’s head typically scans the scene left to right and back during the course of natural walking. The camera looks at same scene multiple times, thus forming a series of short local visual loop closures. We take advantage of this phenomenon to improve the accuracy of the estimated camera poses.

We maintain an archive of the past keyframes, which is continuously updated whenever a new keyframe is initialized. An incoming frame I' is not only mapped to the current keyframe I but also to the previous keyframes $I_{t-1}, I_{t-2} \dots$ for establishing redundant paths. Initially, when a new frame is received, an attempt is made to find a match with an archived keyframe. A match is said to exist when the KL Divergence [27], computed between the histograms of two query images, is small and the estimated camera view vectors point to the same direction. This actually puts a constraint on the current frame that it is looking in the same direction or view as of the keyframe un-

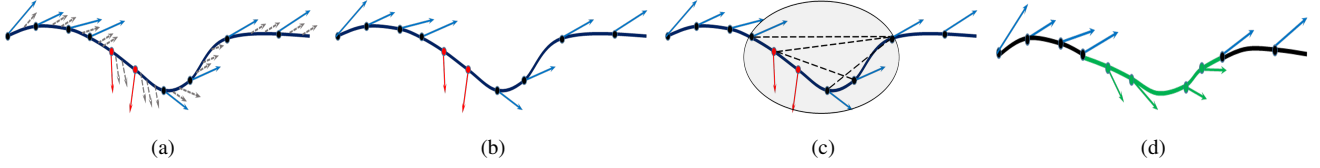


Figure 3. Our approach to correction of poses using short loop closures and rotation averaging. (a) shows a typical trajectory where each camera is connected to the next, and due to large rotation two cameras (indicated in red) are wrongly estimated. The dotted arrows represent intermediate frames (b) only keyframes take part in loop closures (c) shows short loop closure detection using KL divergence and extra matches (dotted edges) are found to close the loops (d) shows the corrected trajectory (green) after running rotation averaging and one iteration of Gauss-Newton for correcting camera positions. Images are best viewed in colour.

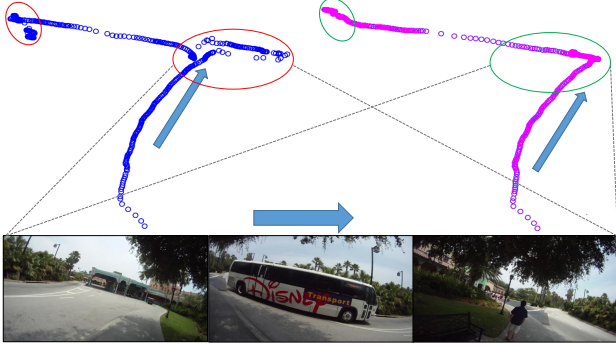


Figure 4. Effect of rotation averaging and Gauss-Newton re-initialization is shown on the Georgia Tech. Social Interaction data set [28]. Trajectory in blue is before refinement with red ovals depicting breaks in the trajectory due to large rotations (some sample images are also shown), which gets corrected after refinement by our method in pink. The trajectory is depicted for a sequence of 500 frames on which we detected multiple batches of short local loop closures within a time frame of 5-10 seconds. Images are best viewed in colour.

der consideration. This constitutes a local loop closure. In our experiments, we select a key frame for every 10 frames. To keep the loop closures local, only the neighbouring keyframes are searched up to a fixed time interval T . Additionally the loop closures are detected on sliding windows for imposing a smoothness constraint over the trajectory.

4.2. Rotation averaging based pose refinement

Short local loop closures give rise to a number of redundant paths between the keyframes. These additional constraints are then used to refine the initial pose estimate. Sharp rotations over a longer period cause the rotation estimates to drift. This drift is corrected by rotation averaging (see section 3.3 for more details). Rotation averaging is performed in small sized windows to ensure enough flexibility for errors to be averaged out. This continuous cycle of pose estimation followed by rotation averaging over a short loop closure ensures that consistent and accurate estimations of the prior camera pose are used to initialize the next set of

incoming frames. **Given the tracking challenges and noisy 3D estimates from an egocentric video, the main source of error in egocentric camera pose estimation is the reliance on these noisy 3D estimates. We do a rotation stabilization using 2D, image based, pairwise rotation estimates, which is not effected by such noisy 3d estimates.**

4.3. Gauss-Newton re-initialization

Although rotation averaging gives robust estimates of camera rotations, it does not alter the camera translations. However, the new rotation estimates are likely to change the camera center estimates. The new camera centers are then re-estimated by running a second level of Gauss-Newton optimization using the newly averaged rotations and previously calculated translations as initial estimates. A single iteration at the highest resolution of the image-pyramid ensures that the rotation estimates remain nearly unchanged and only minor readjustments are brought about in the translations, thereby keeping the final pose estimates restricted in the local neighborhood/minima. This can be achieved at minimal computational cost. **Moreover, this approach is more suitable than simply fixing rotation and updating the translation estimates without 3D refinement, which may make the estimates incompatible and erroneous.**

Fig.4 shows a result from the proposed depicting advantage of short loop closure followed by rotation averaging.

5. Experiments

Robust and accurate camera pose estimation can effectively be used in numerous egocentric applications. However, state-of-the-art visual SLAM solutions such as [5] fail to find accurate trajectories for egocentric videos. This has led to various researchers adapting alternate, sub-optimal approaches in egocentric video based problems [1, 3, 2, 4]. In this section, we focus on several of these problems. We run the proposed method on the datasets associated with each of these problems. However, instead of bypassing egomotion estimation as has been done in the original solutions to these problems, we use the egomotion computed from our algorithm for the tasks at hand. We show that for all the problems that we tested our approach on, the usage of our



Figure 5. Comparison of trajectories obtained for egocentric videos. In each of these images the trajectories obtained by our method are marked in blue, while those attained by LSD-SLAM are marked in red. Due to the lack of any standardized datasets with ground truth poses for egocentric videos, the accuracy of the trajectories is demonstrated by overlaying them on GPS data (pink). It should be noted that the LSD-SLAM trajectories shown here are the ones obtained just before the algorithm crashed. The frequent crossed trajectories in LSD-SLAM results is due to noisy 3D estimates maintained by the algorithm which often leads to false matches

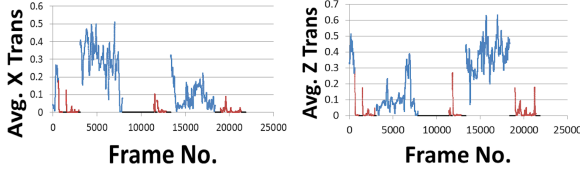


Figure 6. We show the egomotion computed using our technique on a video used in [4, 3]. The left and right figures show the computed X and Z translations respectively. The red curve in the plot shows the frames classified as stationary and the blue curve indicates transit frames. Black curve indicates unused frames in ours as well as the original paper

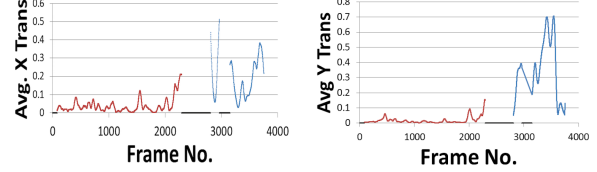


Figure 7. We use our algorithm to compute egomotion in a video from [4] to detect ‘static’ and ‘stair-climbing’ activities. Left and right images show X and Y translation components of the computed egomotion respectively. The red curve indicate frames classified as stationary, blue indicate stair-climbing and black show frames not used in the original as well as our classification.

method results in either similar or improved performance.

We have implemented our version of LSD-SLAM in C++, including short loop closures and the other heuristics mentioned in the last section. All the experiments have been carried out on a regular desktop with Core i7 2.3 GHz processor (containing 4 cores) and 16 GB RAM, running Ubuntu 14.04. Unoptimized implementation of the proposed algorithms runs at 2 to 3 frames per second on such a machine. We will release the source code of our implementation after the acceptance of the paper.

The proposed algorithm requires intrinsic parameters of the camera for egomotion estimation. For the sequences captured by us using GoPro Hero 3+ camera, we have calibrated the camera and used the estimated parameters. For the sequences taken from other sources, we do not have the information about the make and the version of the cameras used. Therefore for all these videos, we used the same set of calibration values as estimated by us for our GoPro Hero 3+. The approximate values appear to work reasonably well in our experiments, indicating the robustness of the proposed algorithm against camera calibration and distortion parameters. We would like to mention that we also tried auto-calibration. However, owing to noisy and inaccurate depth estimates, the calibration estimation never converges. Most of the SLAM techniques that use self-calibration (like [29]) do so in controlled environments. In the absence of exact calibration, we have used standard parameters for GoPro cameras as mentioned earlier.

In the egocentric videos, rotations form the major component of motion because of quick head movements (in comparison to typical slow forward translations). Rotation estimation accuracy is therefore a major requirement in such egomotion estimation problems. We experimented with rotation computations in both quaternion and Lie-algebraic frameworks. We found the Lie-algebra based computations to be more suitable as compared to the quaternion approach which often led to instabilities. The global loop closure techniques used in current state of the art, including LSD, ORB and DT SLAM, do not explicitly make use of the large number of redundant paths that are available in egocentric videos due to to-and-fro head movement. Also most of them rely on the noisy 3D estimates for loop closures. We would like to note that both the contributions (loop closure and rotation averaging using 2D estimates) are essential for the robust camera pose estimation. In our experiments, LSD SLAM with loop closures over short windows also fails due to the usage of noisy 3D estimates.

Since we do not have egocentric videos with ground truth poses (as described earlier, shaky handheld is not equivalent to egocentric), we have relied more on macro level accuracy cues such as overlaying the computed paths on maps (see Fig. 5). We also synchronise the trajectories with the videos to manually verify if the significant events such as wearer turning are captured correctly. It may be noted that the observed instability of the LSD SLAM on the videos shown in Fig. 5 can not be attributed to calibration param-

ters as these were the sequences captured by us and exactly computed calibration parameters have been used for this experiment.

5.1. Temporal Segmentation [3]

Poleg et al. [3] have suggested a technique for temporally segmenting egocentric videos by labeling each frame as one of seven activities. The set of activities chosen by them forms a partition implying that at any point of time, a wearer must be in one of the seven activity states. Many of these activities such as walking, sitting etc. could have been classified by simple egomotion computation. However, they have reported the failure of pose computation on their videos and have suggested optic flow features as an alternative. To test the efficacy of our algorithm, we picked a random video sequence from their dataset, *Huji_Yair.8_part1*. **LSD-SLAM fails on this sequence in consonance with the observation by Poleg et al. However our algorithm works flawlessly. We classify frames into stationary and transit by computing inter-frame X and Z translations² and accumulating them over windows of 10 frames. A SVM classifier is then trained over these features. We obtain an accuracy of 97% for the two class classification problem against the reported accuracy of 98% with the original flow based method.** Figure 6 shows the result.

5.2. Activity Classification [4]

In an another paper of theirs, Poleg et al. [4] have extended the optical flow based technique presented in their original paper [3]. Here, instead of temporal segmentation, they show how to train a compact CNN for classifying long term activities of the wearer based on the suggested sparse optical flow. **While optical flow, can be computed reliably, it is an ambiguous** cue for the considered egocentric activities. Camera egomotion is a much better indicator (if possible to compute reliably) and can improve the quality of prediction. However, as in their original work, Poleg et al. report the failure of SLAM techniques. To show the robustness of our algorithm, we used our algorithm to compute egomotion on one of their videos, “2013 American Lung Association Fight For Air stair climb”. **We then used the computed X and Y translation components of egomotion to distinguish between two classes, stationary vs stair climbing. Similar to the approach described in section 5.1, we accumulate these components over windows of 10 frames thereafter training a SVM classifier over it. We achieved an F1-score of 0.99 for both the classes whereas [4] achieved an F1-score of 0.74 and 0.61 for stair climbing and static respectively. Figure 7 shows the results.** It may be noted that the purpose of the experiments is to show the usefulness of our technique for solving range of problems in variety of video capture

²In the 3D world coordinates, variations in depth, height and left & right directions are marked on Z, Y and X axis respectively.



Figure 8. Comparing proposed approach with naive 10 $\times$ fast forwarding and Egosampling [1] on a publicly available video [2]. The first row shows output from uniform sampling. The second and third rows show outputs from EgoSampling and proposed approach. Focusing on the location of tree and the pedestrian reveals that both EgoSampling as well as proposed approach achieve equivalent stabilization which is much better compared to naive uniform sampling.



Figure 9. Wearer’s gaze fixation is easy to detect in an egocentric video by looking at the constancy of camera look at point. However, Poleg et al. reported failure of egomotion computation and suggested a flow based technique. We use the computed egomotion from our technique on their videos and successfully detect the gaze fixation instances. The figures show some fixations detected by our method.



Figure 10. We tested the proposed method on long shaky sequences given in Hyperlapse [2]. **The trajectory shown is for video sequence gl02.mp4 in Hyperlapse dataset.**

settings. The selection of test sequence/classes is only indicative and random.

5.3. Egosampling [1]

Egocentric videos tend to be long and shaky making them hard to watch. Fast forwarding is a natural solution to quickly browse such videos. However, fast forward by uniform sampling may accentuate the shake already present in

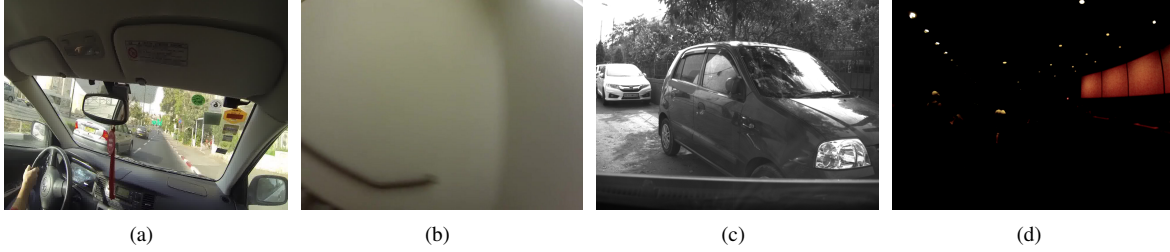


Figure 11. Our algorithm fails in the above cases. In (a) the wearer is part of multiple environments and (b) suffers from extreme blur. In (c) the scene is formed of non Lambertian surfaces causing inaccuracies in the pose estimates. (d) has negligible illumination

egocentric videos. EgoSampling by Poleg et al. is an adaptive frame sampling technique which models frame sampling and video stabilization as a joint optimization problem. The problem is posed as one of finding a shortest path in a graph where each node represents a frame in the video and the edge between frames t and k represents the perceived stability if frame k is placed immediately after frame t in the output. Amongst many costs, one of the costs suggested by Poleg et al. is the ‘shakiness’ cost which biases the selection in favor of forward or similarly looking frames. They have reported failure of SLAM and have used focus of expansion calculated from optical flow for the purpose. We consider the same framework as EgoSampling [1] but use our computed egomotion to compute the shakiness cost. Figure 8 shows that using our method one can achieve similar improvement in stabilization as EgoSampling compared to uniform sampling, but in a more principled manner.

5.4. Gaze fixation [3]

Instead of continuous rotations of the camera due to the constant head movement of the wearer, egocentric videos sometime exhibit anomalies when this natural head rotation temporarily ceases. The anomalies usually indicate wearer’s gaze fixation. Poleg et al. have given a flow based technique to detect such gaze fixations [3]. We use our technique to find the egomotion and estimate the camera view vector, corresponding to the direction of the wearer’s gaze. **A gaze fixation can then be detected by looking at the constancy of the camera look at point.** As in the original paper, we use a moving window of 5 seconds to detect gaze fixation. Our hypothesis is based on identifying a point of intersection of the view vectors for the frames within this moving window. We use the proposed approach to identify gaze fixation points in two videos from the “HUJI EgoSeg dataset (Huji_Chetan_1 and Huji_Yair_2)”. The gaze annotation was not available in the dataset and we manually annotated the frames. From the 17 gaze fixation points, our method successfully detected 16 fixations with 3 false positives, which is comparable to what has been reported in the original work.

5.5. Hyperlapse [2]

Hyperlapse gives a technique to smoothly fast-forward a long hand-held or egocentric video with rapid camera motion [2]. The algorithm uses a structure from motion technique [30] to compute camera trajectory as well as the 3D map. Since the SfM algorithm do not scale well the videos were divided in batches of 1400 frames. We tested our algorithm as well as LSD-SLAM on a video from their dataset. Figure 10 shows the result. While LSD-SLAM fails, our method works even on long sequences of 10,000 frames.

5.6. Failure Cases

Our experiments indicate that the proposed algorithm is much more robust compared to state of the art algorithms like LSD-SLAM. However, we could identify multiple cases where the work still needs improvement. Figure 11 shows some failure cases.

6. Conclusion

Egomotion is an important information in many egocentric applications. However, its use has been restricted until now because of failure of state-of-the-art visual SLAM techniques on such videos in the wild. We exploit the typical motion profile in an egocentric video to perform local loop closures based on realignment of wearer’s gaze directions. This allows us to fix the camera pose locally and efficiently. Noting that camera motion is dominant 3D rotation, we use two step loop closure doing rotation averaging first followed by translation fixation. Our experiments indicate that exploiting these properties of egocentric videos leads to a robust camera pose estimation algorithm. Our experiments on many egocentric applications where egomotion has been reported to fail indicates that the proposed algorithm can be used successfully. This resolves a long standing problem in egocentric vision unlocking the use of egomotion in many egocentric applications.

References

- [1] Y. Poleg, T. Halperin, C. Arora, and S. Peleg, “EgoSampling: Fast-forward and stereo for egocen-

- tric videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4768–4776.
- [2] J. Kopf, M. F. Cohen, and R. Szeliski, “First-person hyper-lapse videos,” *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 78:1–78:10, 2014.
- [3] Y. Poley, C. Arora, and S. Peleg, “Temporal segmentation of egocentric videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2537–2544.
- [4] Y. Poley, A. Ephrat, S. Peleg, and C. Arora, “Compact CNN for indexing egocentric videos,” in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016.
- [5] J. Engel, T. Schops, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 834–849.
- [6] Google, “Glass,” <https://www.google.com/glass/start>. [Online]. Available: <https://www.google.com/glass/start>
- [7] GoPro, “Hero3,” <https://gopro.com/>. [Online]. Available: <https://gopro.com/>
- [8] T. Starner, J. Weaver, and A. Pentland, “Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 20, no. 12, pp. 1371–1375, 1998.
- [9] H. Pirsavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2847–2854.
- [10] A. Fathi, A. Farhadi, and J. M. Rehg, “Understanding egocentric activities,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 407–414.
- [11] S. Sundaram and W. Mayol-Cuevas, “High level activity recognition using low resolution wearable vision,” in *Proceedings of the IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2009, pp. 25–32.
- [12] M. S. Ryoo and L. Matthies, “First-person activity recognition: What are they doing to me?” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2730–2737.
- [13] Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2714–2721.
- [14] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, “Fast unsupervised ego-action learning for first-person sports videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3241–3248.
- [15] Y. J. Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1346–1353.
- [16] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3748–3754.
- [17] J. Engel, J. Sturm, and D. Cremers, “Semi-dense visual odometry for a monocular camera,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1449–1456.
- [18] G. Klein and D. Murray, “Parallel tracking and mapping on a camera phone,” in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009, pp. 83–86.
- [19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “DTAM: Dense tracking and mapping in real-time,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2320–2327.
- [20] A. Chatterjee and V. M. Govindu, “Efficient and robust large-scale rotation averaging,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 521–528.
- [21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [22] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, “A comparison of loop closing techniques in monocular SLAM,” *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1188–1197, 2009.
- [23] L. Clemente, A. Davison, I. Reid, J. Neira, and J. D. Tardós, “Mapping large loops with a single hand-held camera,” in *Proceedings of Robotics: Science and Systems Conference*, 2007.

- [24] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [25] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. D. Tardos, "An image-to-map loop closing method for monocular SLAM," in *Proceeding of the International Conference on Intelligent Robots and Systems*, 2008, pp. 2053–2059.
- [26] M. J. Leotta, E. Smith, M. Dawkins, and P. Tunison, "Open source structure-from-motion for aerial video," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.
- [27] P. J. Moreno, P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for svm classification in multimedia applications," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2004, pp. 1385–1392.
- [28] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 1226–1233.
- [29] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. S. Kweon, "High-quality depth from uncalibrated small motion clip," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [30] C. Wu, "Towards linear-time incremental structure from motion," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2013, pp. 127–134.