University of Dundee

Egomap

Suveges, Tamas; McKenna, Stephen

Published in:

Pattern Recognition. ICPR International Workshops and Challenges

DOI:

[10.1007/978-3-030-68796-0_25](https://doi.org/10.1007/978-3-030-68796-0_25)

Publication date:

2021

Document Version

Peer reviewed version

[Link to publication in Discovery Research Portal](#)

Citation for published version (APA):

Suveges, T., & McKenna, S. (2021). Egomap: Hierarchical First-person Semantic Mapping. In A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, & R. Vezzani (Eds.), Pattern Recognition. ICPR International Workshops and Challenges : Virtual Event, January 10–15, 2021, Proceedings, Part III (pp. 348-363). (Lecture Notes in Computer Science; Vol. 12663). Springer . https://doi.org/10.1007/978-3-030-68796-0_25

General rights

Copyright and moral rights for the publications made accessible in Discovery Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from Discovery Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Egomap: Hierarchical First-person Semantic Mapping^{*}

Tamas Suveges^[0000–0003–0737–7348] and
Stephen McKenna^{**[0000–0003–0530–2035]}

CVIP, School of Science and Engineering,
University of Dundee, Dundee DD1 4HN, U.K.
`t.suveges@dundee.ac.uk`; `s.j.z.mckenna@dundee.ac.uk`

Abstract. We consider unsupervised learning of semantic, user-specific maps from first-person video. The task we address can be thought of as a semantic, non-geometric form of simultaneous localisation and mapping, differing in significant ways from formulations typical in robotics. Locations, termed *stations*, typically correspond to rooms or areas in which a user spends time, places to which they might refer in spoken conversation. Our maps are modeled as a hierarchy of probabilistic station graphs and view graphs. View graphs capture an aspect of user behaviour within stations. Visits are temporally segmented based on qualitative visual motion and used to update the map, either by updating an existing map station or adding a new map station. We contribute a labelled dataset suitable for evaluation of this novel SLAM task. Experiments compare mapping performance with and without the use of view graphs and demonstrate better online mapping than when using offline clustering.

1 Introduction

First-person (egocentric) video, acquired using wearable cameras, provides data streams that lend themselves to inference and learning about everyday activities from a human-centred perspective. They have been used to analyse social interaction, focus of attention, and lifestyle behaviour, and find applications in assistive technology, health, and human-robot interaction, for example. This paper explores the use of first-person video to automatically construct a representation of a user’s everyday environment in terms of locations that they frequent and their patterns of transition between, and their behaviours within, those locations. The locations in our model are conceived of as places where the user stays for some time, often to perform a specific activity or task. They tend to correspond to places to which people might refer in narrative, e.g., my desk,

^{*} This research received funding from the UK Engineering and Physical Sciences Research Council (EPSRC grant EP/N014278/1). Maxwell J. Wilson assisted with video data acquisition. We are grateful to members of the CVIP cluster and the ACE-LP project for helpful discussions and feedback.

^{**} Corresponding author

the kitchen, the meeting room. We will call these locations *stations*. The aim is to dynamically learn a semantic map in terms of stations that can be used to localise the user and provide contextual information from previous visits to those stations.

As a motivating example application, consider voice output communication aids for non-speaking users. Their users often average 8-12 spoken words per minute; typical adult speech rates are 150-190 words per minute. One approach to offering a user better predictive text, and thus improved communication speed, is to leverage information about the context [1]. A user-specific map would enable that user’s communication to be contextualised in terms of the stations they repeatedly visit. Other applications could include aids to compensate for memory loss and automatic visual personal history generation, potentially combined with activity recognition [2–4].

The task that we address can be thought of as a semantic, non-geometric, human-centred form of simultaneous localisation and mapping (SLAM). This is done without recourse to metric map building which is not necessary to address this task. We propose a hierarchical probabilistic model, incorporating station graphs and view graphs, and hypothesise that it can be used to perform the unsupervised mapping and localisation that we require. Each view graph captures aspects of the user’s head movements while at a station. A characteristic of our map representation is that two stations can be adjacent in the station graph even if one is physically only indirectly reachable from the other; map adjacency does not imply spatial adjacency of stations. This is because the user can transition through intervening space without stopping. We identify transitions between stations based on qualitative visual motion analysis. Maps are dynamically updated, and new stations are added as they are visited for the first time.

In common with many SLAM systems, we adopt a visual bag of words feature representation with traditional features. Deep convolutional neural network representations have been used in SLAM but can perform poorly if not configured carefully for the task [5]. We leave the exploration of the use of different view representations within our framework, including the use of deep features, to future work.

The main contributions are as follows.

1. We formulate a novel first-person mapping and localisation task which requires unsupervised, sequential learning of a user-centred map in terms of semantic stations.
2. We propose a method to perform this task based on a hierarchical, probabilistic model and we report results comparing learned maps with ground-truth maps, including on a new egocentric dataset which we make available.

2 Relationship with Previous Work

In this paper, we rely solely on egocentric video sensing from a single head-mounted camera, and we focus on semantic (non-geometric) visual mapping and localisation.

2.1 Related approaches

Appearance-based SLAM methods use visual data to construct topological maps, often employing omnidirectional cameras to ease the problem of dealing with different views of the same place [6–8]. For example, a modified incremental spectral clustering method was proposed to cluster visually similar places based on SIFT features [6]. The performance of different image descriptors has been assessed in such a setting using manually taken omnidirectional images [7]. In contrast, first-person video is usually acquired using a monocular camera with a relatively limited field of view.

Three-dimensional reconstruction is another approach to detect global loop closures, borrowed from robotics and often requiring expensive computation. Ego-Slam [9] highlights the importance of local loop closures in an egocentric video to handle significant rotation, e.g., head movements. Detected local loop closures are used to improve the accuracy of estimated camera poses, resulting in more accurate 3D reconstruction. In contrast, we use such information to build a graph representation of a station in which edges and nodes store information on head movement and visual appearance, respectively.

Furnari et al. [10, 11] and Ortis et al. [12] formulate a supervised classification task rather than unsupervised SLAM. In common with [13], these are off-line solutions; we address the more challenging problem of developing an unsupervised method that can operate online.

SeqSLAM compares sequences of observations to detect loop closures [14]. First-person videos are, by their nature, generally long, so storing them and performing SeqSLAM on them in their entirety is impractical, especially as computation time exponentially increases with the length of visits.

FabMap implements a probabilistic framework for SLAM using Bayesian filtering [15]. When a loop closure is detected, FabMap updates the appearance model of a given place. This method can work well when the environment is captured at low frame-rate from a moving robot such that consecutive images are visually dissimilar enough (Consecutive images in the dataset used in [15] were acquired at positions separated by $1.5m$). Otherwise, the visual smoothness of the environment over time can cause FabMap to update a single appearance model continuously, necessitating the use of keyframe extraction in practice. Furthermore, FabMap, similarly to many other SLAM approaches, does not represent the hierarchical structure of the environment, instead defining a location as a single image.

EgoMap, the probabilistic framework we propose, uses Bayesian filtering for map inference and learning similarly to FabMap [15] but offers a more informative, hierarchical representation of the environment. We conceptualise a station

as a place where the user spends time. Each station is then described as a graph of interconnected views between which the edge weights represent the probability of transition from one view to another. Hence, EgoMap differs from existing SLAM techniques by conceptualising stations, resulting in a human-centric representation. Furthermore, EgoMap offers a hierarchical representation of the user’s world that allows the framework to detect loop closures at a higher, hence more general, level. Most SLAM techniques only allow image-level loop closure detection, which is useful in robotics but misses the semantic interpretation in terms of places.

2.2 Datasets

We address a novel SLAM challenge and its empirical evaluation accordingly requires an appropriate first-person video dataset. This dataset should capture the repetitive and somewhat habitual nature of everyday behaviour in which visits are made repeatedly to certain stations, often to perform specific actions. Visits to specific stations have associated user-specific visual behaviours depending on the purpose of the visit.

SLAM datasets such as KITTI 06, City Center and KAIST [16] are recorded from road vehicles often moving at relatively constant speeds. These datasets were created for a different purpose and fail to capture repetitive human behaviour in a personal environment. Similarly, those captured by robots, such as NewCollege and Robot@Home [17], focus on capturing the environment uniformly while continuously moving through it, and they fail to record everyday activity in which a person transitions between locations in a somewhat habitual manner. SLAM datasets do not capture such behaviour so are unsuitable for our purposes [18–21]. An absence of loop-closures is also common with SLAM datasets given that the environment is only captured once. Visit level loop-closures are essential to evaluate unsupervised semantic mapping. Furthermore, datasets like [19–21] have short video segments omitting the transitional sections between stations.

Many egocentric datasets focus on activity recognition [22–26]. Datasets tailored for such a task often include short video segments capturing multiple activity sequences each recorded at different stations. This setup leads to problems such as missing transitional sections and lack of loop-closures similarly to those in SLAM. Lifelogging is another intense area in ubiquitous computing [3, 4, 2], however, datasets are typically low framerate [4, 2, 27–29]. Those captured at high frame rate commonly document the user’s life for a short period of time resulting in only a few or no loop-closures [30, 31]. There are several other impressive egocentric datasets, but the number of loop closures is limited to one per station which is insufficient for our purposes [10–12]. EGO-CH [32] is an extensive egocentric video dataset for cultural heritage visitor behavioural understanding recorded by 70 different people exploring museums. This dataset is not a good match for our purposes as locations are both visually and semantically very similar. We aim to map the user in their everyday environment where stations differ in their utilization. A more relevant dataset records the everyday

Fig. 1: (a) An illustrative map with five station nodes. Edge weights are transition probabilities. (b) Example images from four different stations in our dataset; from top left to bottom right: 3D-lab, Coffee-area, Kitchen 1, and Lab. (c) Views learned for Kitchen 1

life of 20 subjects in their own environment using a chest-mounted camera [2]. However, due to a bias towards activity recognition, many of the users switch off the camera between activities, leading to missing transitions. Furthermore, the number of stations is often rather small (3-4). Nonetheless, we report additional, preliminary results on two of the videos from that dataset.

We contribute a dataset more suited to the challenge addressed in this paper. It includes transition segments between stations, multiple visits by a user to multiple distinct stations, and unique labels for *all* visited stations.

3 Model

3.1 Hierarchical map

We introduce a hierarchical map structure in which stations in a map are represented in terms of view graphs. Our map can be written as a weighted directed graph $\mathcal{M} = (\{\mathcal{L}_1, \dots, \mathcal{L}_M\}, \mathbf{E})$ where M is the number of stations in the map and $\mathbf{E}$ is an $M \times M$ stochastic transition matrix with $\mathbf{E}_{i,j} = P(l_{t+1} = j, |l_t = i)$ denoting the probability of transitioning from station i at time t to station j at time $t + 1$. Each station is in turn represented as a directed graph $\mathcal{L}_m = (\{\mathcal{V}_1, \dots, \mathcal{V}_{N(m)}\}, \mathbf{P})$ where $N(m)$ is the number of views at that station and $\mathbf{P}$ is an $N(m) \times N(m)$ stochastic transition matrix with $\mathbf{P}_{i,j} = P(v_{t+1} = j | v_t = i)$ denoting the probability of transitioning from the j^{th} view to the i^{th} view at that station. Transition matrix $\mathbf{E}$ encapsulates information about the user’s movement around the map whereas each $\mathbf{P}$ matrix captures information about user behaviour at a map station. This structure, illustrated in Fig. 1a, allows us to build a personalised ego-centric map for a specific user where behavioral information can be used to aid localization. Fig. 1b shows examples images from four stations, and Fig. 1c shows four example view images from one of those stations.

3.2 View Representation

Each view has an associated appearance model. In our experiments we use a bag-of-words representation [33]. Each view is represented by a vector in which each element indicates the presence or absence of a ‘word’ in a vocabulary of size D . Let $\mathbf{z}_t = [z_1, \dots, z_D]^T$ denote an observation at time t where each z_d is a binary variable. We used quantized SURF descriptors to generate binary features. Rather than model a view directly in terms of which features are likely to be observed, we introduce hidden variables similarly to [15]. A detector model relates feature existence e_d to feature detection z_d . The detector is specified by

$$\begin{cases} P(z_d = 1 | e_d = 0), \text{false positive probability.} \\ P(z_d = 0 | e_d = 1), \text{false negative probability.} \end{cases} \quad (1)$$

Variable e_d can be thought of as encoding whether or not an object that generates observations of type z_d exists. We model the d^{th} view as a D -vector, $\mathbf{v}$, the d^{th} element of which records $P(e_d = 1)$ for that view. Each of the feature generating objects, e_d , is generated independently by the view.

4 Motion Analysis for Transition Detection

We aim to represent the user’s world as a set of interconnected stations. Each station is a place in the world that the user spends time visiting. Such visits are interspersed with periods of transition characterised by, for example, sustained walking. We propose to detect when such periods of transition between visits are occurring based on qualitative visual motion analysis.

We compute frame-to-frame displacement vectors using a Lucas-Kanade feature tracker [34] on a 10×5 grid similarly to [31]. Transitions in egocentric video tend to be dominated by forward motion [9]. We assume that translation along the camera axis can be estimated by a simple zoom [35]. Equations (2) and (3) estimate the new coordinates of a point on the image plane given zoom parameter β and focal length f .

$$X' = f[\tan^{-1} \frac{X}{f}](1 + \frac{X^2}{f^2})\beta \quad (2)$$

$$Y' = f[\tan^{-1} \frac{Y}{f}](1 + \frac{Y^2}{f^2})\beta \quad (3)$$

We substantiate the coordinates of the grid cell centres and find the zoom parameter value, β^* , that gives the best fit to the estimated optical flow. Equations (2) and (3) do not decompose the motion field with respect all camera motion parameters, instead focusing on the zoom parameter. Once estimated, β^* is temporally smoothed to give $\hat{\beta}_t = s\hat{\beta}_{t-1} + (1-s)\beta_t^*$. This smoothed value is then thresholded to segment the video, detecting transition when $\hat{\beta} > T_{transition}$. We set $s = 0.99$ in all our experiments.

5 Inference

5.1 View inference via recursive Bayes

At time t , given an observation sequence $\mathbf{z}_{\tau:t}$ for the current visit to the m^{th} map station, we can recursively infer the discrete random variable v_t over the $N(m)$ views,

$$P(v_t|\mathbf{z}_{\tau:t}) = \frac{P(\mathbf{z}_t|v_t) \sum_{v_{t-1}} P(v_t|v_{t-1})P(v_{t-1}|\mathbf{z}_{\tau:t-1})}{P(\mathbf{z}_t|\mathbf{z}_{\tau:t-1})}, \quad (4)$$

where the visit started at time $\tau < t$, $P(\mathbf{z}_t|v_t)$ is the current view likelihood, $P(v_t|v_{t-1})$ is the transition probability from matrix $\mathbf{P}$, and $P(\mathbf{z}_t|\mathbf{z}_{\tau:t-1})$ is the normalizing term. At the start of a visit, at time τ , a prior distribution is required over the views at a station. It would be possible to obtain this prior using weighted connections between views from different stations. However, for simplicity, we assume independence between views from different stations and use a flat prior that assigns equal probability to each view in a station at the start of a visit:

$$P(v_\tau = j|l = m) = \frac{1}{N(m)} \quad (5)$$

The view likelihood cannot be evaluated directly because of the intractability of learning the high-order conditional dependencies between appearance words. The simplest approximation is naive Bayes,

$$P(\mathbf{z}|v_t) \approx P(z_D|v_t) \dots P(z_2|v_t)P(z_1|v_t), \quad (6)$$

where,

$$P(z_d|v_t) = \sum_{s \in \{0,1\}} P(z_d|e_d = s)P(e_d = s|v_t) \quad (7)$$

5.2 Station inference

During a visit, we can localise the user by estimating a probability distribution over map stations. At time t , given a map $\mathcal{M}$, an observation sequence $\mathbf{z}_{\tau:t}$ for the current visit, and the observation sequences, $\mathbf{Z}$, from all preceding visits, we can infer a discrete random variable, l_t , over stations,

$$P(l_t|\mathbf{z}_{\tau:t}, \mathbf{Z}) = \frac{P(\mathbf{z}_{\tau:t}|l_t) \sum_{l_{t-1}} P(l_t|l_{t-1})P(l_{t-1}|\mathbf{Z})}{P(\mathbf{z}_{\tau:t-1}|\mathbf{Z})} \quad (8)$$

where $P(\mathbf{z}_{\tau:t}|l_t)$ is the station likelihood and $P(l_t|l_{t-1})$ is the transition probability from matrix $\mathbf{E}$. Notice that the summation is constant during a visit so need only be computed once per visit. During the first visit in a video we use a flat prior over map stations:

$$P(l = m | \mathcal{M}) = \frac{1}{M} \quad (9)$$

The likelihood of a station, $P(\mathbf{z}_{\tau:t} | l_t = m)$, can be computed using the forward algorithm. Let us define a forward variable α initialized as $\alpha_\tau(v_\tau) = P(\mathbf{z}_\tau | v_\tau)P(v_\tau)$. Similarly to Equation (4) we get the induction step:

$$\alpha_{t+1}(v_t) = P(\mathbf{z}_t | v_t) \sum_{v_{t-1}} P(v_t | v_{t-1}) \alpha_t(v_{t-1}) \quad (10)$$

The likelihood at any time t' during a visit is given by the termination step:

$$P(\mathbf{z}_{\tau:t'} | l = m) = \sum_{v_{t'}} \alpha_{t'}(v_{t'}) \quad (11)$$

6 Learning

6.1 View update

At each time step t , an association decision is made based on Equation (4) as to which view's appearance model, $\mathbf{v}$, to update. A view model is the mean of all observations associated with it up until the current time, t . The model is sequentially updated as a moving average using Equation (12) where γ is inversely proportional to number of updates already performed on the view. When a new view is created, its appearance model is initialized so that all words exist with marginal probability $P(e_d = 1)$ derived from the training data.

$$\mathbf{v}[d] := \gamma P(e_d = 1 | \mathbf{z}_t) + (1 - \gamma) \mathbf{v}[d] \quad (12)$$

6.2 View creation

In Section 5.1, we described how to compute a probability distribution over views (Equation (4)). However, we wish to dynamically add new views to the model as appropriate. Therefore, we adjust the denominator in Equation (4) so that the probabilities of the known views sum to less than one, leaving some probability mass assigned elsewhere. Conceptually, we would like to add a sum over all the yet to be modelled views, $\bar{v}_t$, at the station:

$$\begin{aligned} P(\mathbf{z}_t | \mathbf{z}_{\tau:t-1}) = & \sum_{v_t} P(\mathbf{z}_t | v_t) \sum_{v_{t-1}} P(v_t | v_{t-1}) P(v_{t-1} | \mathbf{z}_{\tau:t-1}) + \\ & \sum_{\bar{v}_t} P(\mathbf{z}_t | \bar{v}_t) \sum_{v_{t-1}} P(\bar{v}_t | v_{t-1}) P(v_{t-1} | \mathbf{z}_{\tau:t-1}) \end{aligned} \quad (13)$$

The second summation cannot be evaluated so we adopt a mean field approximation, similarly to [15]:

$$P(\mathbf{z}_t|\mathbf{z}_{\tau:t-1}) = \sum_{v_t} P(\mathbf{z}_t|v_t) \sum_{v_{t-1}} P(v_t|v_{t-1})P(v_{t-1}|\mathbf{z}_{\tau:t-1}) + \\ P(\mathbf{z}_t|v'_t) \sum_{v_{t-1}} P(v'_t|v_{t-1})P(v_{t-1}|\mathbf{z}_{\tau:t-1}) \quad (14)$$

Here v'_t is the average view. We adopt a uniform prior in the second term, with $P(v'_t|v_{t-1})$ the same for all views.

6.3 Station update

To update a station, we first make a decision about the station of the user on the map, $\mathcal{M}$. To do so we calculate a MAP estimate, l_t^* :

$$l_t^* = \arg \max_m P(l = m|\mathbf{z}_{\tau:t}, \mathbf{Z}, \mathcal{M}) \quad (15)$$

After localization, we update the station's mean field $\mathbf{v}'$ as:

$$\mathbf{v}'[d] := \gamma \frac{1}{(t - \tau) + 1} \sum_{t'=\tau}^t P(e_d = 1|\mathbf{z}_{t'}) + (1 - \gamma)\mathbf{v}'[d] \quad (16)$$

where $\mathbf{v}'$ is the view model of the new mean-field estimate, and γ is the weighting term inversly proportional to the number of examples already used to estimate the mean-field. Once the new mean-field is computed, we use $\mathbf{z}_{\tau:t}$ to iteratively update the station. At each timestep t' , we infer $P(v_t|\mathbf{z}_{\tau:t'})$ and make a decision on the user's view in the station using Equation (4) after which we update the station in accordance with Sections 6.2 and 6.3

6.4 Station creation

Similarly to the mechanism for handling new views, we would ideally adjust the denominator of Equation (8) to be

$$P(\mathbf{z}_{\tau:t}|\mathbf{Z}) = \sum_{l_t} P(\mathbf{z}_{\tau:t}|l_t) \sum_{l_{t-1}} P(l_t|l_{t-1})P(l_{t-1}|\mathbf{Z}) + \\ \sum_{\bar{l}_t} P(\mathbf{z}_{\tau:t}|\bar{l}_t) \sum_{l_{t-1}} P(\bar{l}_t|l_{t-1})P(l_{t-1}|\mathbf{Z}), \quad (17)$$

where $\bar{l}_t$ indexes over the yet to be modelled stations. Although, a mean-field estimation is possible it is not straightforward to estimate it. We instead use a likelihood threshold, $T_{new\ station}$, to decide when to create a new map station.

$$T_{new\ station} = \left[\sum_{\bar{l}_t} P(\mathbf{z}_{\tau:t}|\bar{l}_t) \right]^{\frac{1}{(t-\tau)+1}} \quad (18)$$

$P(\mathbf{z}_{\tau:t}|l_t)$ is computed using a recursive forward algorithm; its value is exponentially dependent on the number of observations. $T_{new\ station}$ can be interpreted

as the average observation likelihood yielded by all the unknown stations for a single observation $\mathbf{z}_t$.

6.5 Transition matrix updates

Transition matrices $\mathbf{E}$ and $\mathbf{P}$ are updated using MAP estimation with a uniform prior over transitions. This avoids transition probabilities of zero. We implement this by initialising all transitions with a count of 5 in our experiments. Each time a transition is observed its count is incremented. Matrices $\mathbf{E}$ and $\mathbf{P}$ are obtained from these count matrices by normalisation. The effect of the prior diminishes over time.

7 Experiments and Metrics

7.1 Dataset

We recorded four hours of video, over five mornings, of a user’s activity on the university campus. This activity was loosely scripted in order to ensure multiple visits were made to a range of stations. Scripts consisted of lists of stations along with suggestions for activities during station visits, e.g., make a drink, read something, collect an item from the printer. A ground-truth temporal segmentation of each video into visit and transition segments was manually annotated with reference to the scripts. The annotator was instructed to mark the start of a transition as the frame when the user appears to intentionally start moving away from the current station, and the start of a visit as the frame at which such transitional movement ends. All five videos were recorded using a head-mounted GoPro Hero 4 at 1080p resolution and 25fps. The head mount gave us the ability to capture the approximate viewpoint of the user.

7.2 Dictionary Learning and Parameter Tuning

We used the first 10,000 place images from the Places dataset to train our dictionary [36]. We extracted 128-dimensional SURF descriptors from each image and trained k -means on the extracted features ($k = 10,000$ as in [15]). To describe an image, we first extract SURF features, quantise using the k clusters, and construct a histogram. The final descriptor is then a binary quantized version of the obtained histogram where each bin implies presence or absence of a feature-generating object (see Section 3.2). We used the first video in our dataset (video 1) to tune the threshold parameter for the transitional motion detector which was then fixed ($T_{transition} = 1.008$).

7.3 Map evaluation

To evaluate mapping performance, we aim to measure how well the predicted map approximates the ground truth map. First, predicted visit segments are

matched with all ground-truth segments with which they overlap (in time). A matrix is then constructed by recording the co-occurrence frequencies between the set of ground-truth visit labels and the set of learned visit labels. Once normalised this can be considered as a joint probability distribution $P(G, L)$ where G and L are random variables over the ground-truth and the learned stations, respectively.

A measure of map quality can be obtained by computing the uncertainty a one-to-one matching process between the possible values in G and L would involve. In a perfect system, nodes co-occur exclusively, and the matching process involves no uncertainty. In other words, knowledge of one random variable provides maximum information on the other. Drifting away from the ideal case, the exclusivity disappears, and the matching process becomes difficult. To evaluate performance, we measure the mutual information (MI) as in Equation (19) where H is the entropy. Mutual information can be interpreted as a measure of how well a one-to-one mapping could be established between values in G and L .

$$MI(L, G) = H(L) - H(L|G) \quad (19)$$

However, MI tends to be higher when the node count is higher. This is undesirable since it prevents comparing graphs with different numbers of nodes. Therefore, we use an adjusted version of MI, AMI [37], that has been used to compare clusterings with varying numbers of clusters:

$$AMI(L, G) = \frac{MI(L, G) - E[MI(L, G)]}{\max(H(L), H(G)) - E[MI(L, G)]} \quad (20)$$

Here, E denotes the expected value. AMI takes the value of 1 when one-to-one correspondence can be established between the predicted map and the ground-truth map with certainty. On the other hand, the value is zero when the number of nodes in the predicted map is either one or equal to the number of visits.

7.4 Baseline and Ablation

We obtain baseline results by running k-means on our dataset. During clustering, we represent each visit using its mean feature vector. Notice that by doing so, we introduce the same time constraint used during inference in our system, i.e., we enforce the must-link constraint on frames from the same visit. We wish to emphasise that the offline nature of our baseline algorithm means it can access all features during clustering, greatly easing the complexity of the task. However, we believe that such a general and well-used approach provides a good baseline for comparison.

Additionally, we perform an ablation experiment in which we remove the view-graph to evaluate its effect. We introduce a prohibition such that the system is only allowed to create one view per station. Such a system strongly resembles FabMap [15] with the added time-constraint.

Fig. 2: Visualisations of (a) a ground-truth map, (b) visit matching between learned and ground-truth locations

8 Results

8.1 Visualisation

Fig. 2 presents one way to visualise an example map and its correspondence with ground-truth. The ground-truth map is shown in Fig. 2a. Its nine stations are represented around the perimeter; the internal connections represent transitions that occur between those stations. Fig. 2b shows a learned map and the ground-truth map together. Stations in the learned map appear around the bottom half of the circle. Connections represent the matching of visits to stations on the learned map with ground-truth stations. The size of the representation of each station in the learned map correlates with the number of visits to the station. In this example, some ground-truth stations connect with multiple nodes in the learned map. For example, *3D-lab* is matched with three nodes. This suggests that the threshold $T_{new\ station}$ was too high, causing the model to generate too many stations. On the other hand, one learned node is well-connected with both *Lab* and *3D-lab*. From Fig. 1b, we can see that these two stations have strong visual aliasing due to a computer monitor.

8.2 Quantitative evaluation

$T_{new\ station}$ is an important free parameter in our system which indirectly controls the final station count. We report performance for different values allowing us to compare predicted maps with varying numbers of stations. In the case of k-means, the number of clusters is directly controllable hence we obtain high-resolution baseline results. We report AMI scores of three methods on our dataset in Fig. 3(a). Our partial system (without view graphs) has similar performance

Fig. 3: AMI using (a) automatic and (b) manual segmentation

to the baseline k-means. However, it is important to highlight that while k-means operates in an offline manner, our partial system is still online. Furthermore, we conclude that the view graphs do have a positive effect on performance. Interestingly, we see that the best performing maps are over-segmented. One could argue that over-segmentation helps overall because ambiguous visits are not mistakenly forced into one of the known stations which would impact the system negatively in the long run given its online nature.

We also report preliminary, partial experiments in Fig. 4 on the ADL dataset published in [2]. Results were acquired by running our algorithm on the 7th and 13th video which have 6 and 3 stations, respectively. These results suggest that the hierarchical representation of the environment tends to help. However, given the low visit counts (6 and 14) and low station counts, those results are noisy.

To ascertain whether the qualitative motion analysis was adversely affecting results, we re-ran the experiment of Fig. 3(a) using manually segmented visits. Fig. 3(b) shows the result. There is little noticeable difference from Fig. 3(a), suggesting that the motion analysis used to detect transitions between visits performs sufficiently well to support map building and semantic localisation.

9 Conclusions

We formulated a new person-centric SLAM task and described a modular, hierarchical, probabilistic framework to tackle it. Unsupervised, sequential learning of semantic, user-specific maps from first-person video enabled us to build a user-specific representation. These maps represent a network of visited stations (locations) and model each station as a network of views. We contributed a dataset and evaluated performance in terms of map correctness.

Transitions between station visits were detected using a qualitative motion analysis method and mapping performance suggested that this worked sufficiently well. Nevertheless, there is room for improvement in other modules in future work. In the view representation and likelihood function we used a quantized

Fig. 4: Preliminary ADL results: (a) video 7 (b) video 13

binary bag-of-visual-words descriptor. Although such descriptors enjoy success in various SLAM systems, they have limitations. Due to the high dimensionality of the binary feature vector, we find that matching likelihoods for two similar and two dissimilar images are too far apart in the log space often resulting in saturated probabilities. Future work could focus on finding representations where the computed likelihoods result in ‘softer’ probabilities. Furthermore, we note that the use of mean-field approximation for views can result in view representations at different stations that are inconsistent at a semantic level. Future work could further explore dynamic creation and merging of view nodes, and of station nodes.

References

1. P.-O. Kristensson, J. Lilley, R. Black, and A. Waller, “A design engineering approach for quantitatively exploring context-aware sentence retrieval for nonspeaking individuals with motor disabilities,” in *Proc. CHI Conference on Human Factors in Computing Systems*, 2020.
2. H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2847–2854.
3. J. Gemmell, C. Bell, and R. Lueder, “MyLifeBits: A personal database for everything,” *Communications of the ACM*, vol. 49, pp. 89–95, 2006.
4. S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Woodberry, “SenseCam: A retrospective memory aid,” in *Proceedings of the 8th International Conference of Ubiquitous Computing (UbiComp 2006)*, 2006.
5. Y. Hou, H. Zhang, and S. Zhou, “Convolutional neural network-based image representation for visual loop closure detection,” in *IEEE International Conference on Information and Automation*, 2015, pp. 2238–2245.
6. C. Valgren, T. Duckett, and A. Lilienthal, “Incremental spectral clustering and its application to topological mapping,” in *Proc IEEE International Conference on Robotics and Automation*, April 2007, pp. 4283–4288.

7. L. Payá, W. Mayol, S. Cebollada, and O. Reinoso, "Compression of topological models and localization using the global appearance of visual information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
8. E. Garcia-Fidalgo and A. Ortiz, "Hierarchical place recognition for topological mapping," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1061–1074, 2017.
9. S. Patra, K. Gupta, F. Ahmad, C. Arora, and S. Banerjee, "EGO-SLAM: A robust monocular SLAM for egocentric videos," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 31–40.
10. A. Furnari, G. M. Farinella, and S. Battiato, "Recognizing personal locations from egocentric videos," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 1, pp. 6–18, 2017.
11. —, "Temporal segmentation of egocentric videos to highlight personal locations of interest," in *Computer Vision: ECCV Workshops*, G. Hua and H. Jégou, Eds., 2016.
12. A. Ortis, G. M. Farinella, V. D'Amico, L. Addesso, G. Torrioni, and S. Battiato, "Organizing egocentric videos of daily living activities," *Pattern Recognition*, vol. 72, pp. 207–218, 2017.
13. Z. Zivkovic, O. Booi, and B. Kröse, "From images to rooms," *Robotic and Autonomous Systems*, vol. 55, no. 5, pp. 411–418, 2007.
14. M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation*, 2012.
15. 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.
16. J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.
17. J. R. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez, "Robot@home, a robotic dataset for semantic mapping of home environments," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 131–141, 2017.
18. D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers, "The TUM VI benchmark for evaluating visual-inertial odometry," in *International Conference on Intelligent Robots and Systems (IROS)*, October 2018.
19. J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2013.
20. W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger, "InteriorNet: Mega-scale multi-sensor photo-realistic indoor scenes dataset," in *British Machine Vision Conference (BMVC)*, 2018.
21. D. Caruso, J. Engel, and D. Cremers, "Large-scale direct SLAM for omnidirectional cameras," in *International Conference on Intelligent Robots and Systems (IROS)*.
22. E. Spera, A. Furnari, S. Battiato, and G. M. Farinella, "EgoCart: a benchmark dataset for large-scale indoor image-based localization in retail stores," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
23. F. Ragusa, A. Furnari, S. Battiato, G. Signorello, and G. M. Farinella, "Egocentric visitors localization in cultural sites," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 12, no. 2, pp. 1–19, 2019.
24. —, "Egocentric point of interest recognition in cultural sites," in *VISIGRAPP (VISAPP)*, 2019.

25. S. Bambach, S. Lee, D. J. Crandall, and C. Yu, “Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions,” in *International Conference on Computer Vision (ICCV)*, December 2015.
26. D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, “Rescaling egocentric vision,” *CoRR*, vol. abs/2006.13256, 2020.
27. M. Aghaei, M. Dimiccoli, C. C. Ferrer, and P. Radeva, “Towards social pattern characterization in egocentric photo-streams,” *Computer Vision and Image Understanding*, vol. 171, pp. 104–117, 2018.
28. E. Talavera, C. Wuerich, N. Petkov, and P. Radeva, “Topic modelling for routine discovery from egocentric photo-streams,” *Pattern Recognition*, vol. 104, p. 107330, 2020.
29. M. Bolaños, Á. Peris, F. Casacuberta, S. Soler, and P. Radeva, “Egocentric video description based on temporally-linked sequences,” *Journal of Visual Communication and Image Representation*, vol. 50, pp. 205–216, 2018.
30. Z. Lu and K. Grauman, “Story-driven summarization for egocentric video,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2714–2721.
31. Y. Poley, C. Arora, and S. Peleg, “Temporal segmentation of egocentric videos,” in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 2537–2544.
32. F. Ragusa, A. Furnari, S. Battiato, G. Signorello, and G. M. Farinella, “EGO-CH: Dataset and fundamental tasks for visitors behavioral understanding using egocentric vision,” *Pattern Recognition Letters*, vol. 131, pp. 150–157, 2020.
33. J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *International Conference on Computer Vision*. IEEE, 2003, pp. 1470–1477.
34. B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *IJCAI*, 1981, pp. 674–679.
35. M. V. Srinivasan, S. Venkatesh, and R. Hosie, “Qualitative estimation of camera motion parameters from video sequences,” *Pattern Recognition*, vol. 30, no. 4, pp. 593–606, 1997.
36. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.
37. N. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Is a correction for chance necessary?” in *International Conference on Machine Learning (ICML)*, 2009.