Automatically and Efficiently Inferring the Hierarchical Structure of Visual Maps

Margarita Chli and Andrew J. Davison
Imperial College London, London SW7 2AZ, UK
{mchli,ajd}@doc.ic.ac.uk

Abstract— **In Simultaneous Localisation and Mapping (SLAM), it is well known that probabilistic filtering approaches which aim to estimate the robot and map state sequentially suffer from poor computational scaling to large map sizes. Various authors have demonstrated that this problem can be mitigated by approximations which treat estimates of features in different parts of a map as conditionally independent, allowing them to be processed separately. When it comes to the choice of how to divide a large map into such 'submaps', straightforward heuristics may be sufficient in maps built using sensors such as laser range-finders with limited range, where a regular grid of submap boundaries performs well. With visual sensing, however, the ideal division of submaps is less clear, since a camera has potentially unlimited range and will often observe spatially distant parts of a scene simultaneously.**

In this paper we present an efficient and generic method for automatically determining a suitable submap division for SLAM maps, and apply this to visual maps built with a single agile camera. We use the mutual information between predicted measurements of features as an absolute measure of correlation, and cluster highly correlated features into groups. Via tree factorisation, we are able to determine not just a single level submap division but a powerful fully hierarchical correlation and clustering structure. Our analysis and experiments reveal particularly interesting structure in visual maps and give pointers to more efficient approximate visual SLAM algorithms.

I. Introduction

As a moving camera (or multi-camera rig) explores its environment, each measurement of the image location of a repeatably observable scene feature provides a probabilistic constraint on its location relative to the camera. It is well understood that many such measurements captured over a long image sequence, in combination with the assumption that most elements of the scene are static, suffice to permit stable estimates of the camera's 3D trajectory as well as a 3D map of the locations of the observed features. The most accurate solution to this estimation problem will be obtained by a batch optimisation approach which seeks the estimates which are most globally consistent with the measurements — a methodology known as bundle adjustment in the photogrammetry and computer vision communities [26], and now generalised by SLAM researchers in graph optimisation frameworks which are able to incorporate all types of sensory input [25], [10].

A. Sparsification for Real-Time Visual Mapping

In robot vision, there has been a natural emphasis on visual localisation and mapping methods which are able to run not as off-line optimisation but as sequential procedures potentially implementable in real-time on modest computing hardware. Real-time operation inevitably requires some form of approximation or sparsification of full global optimisation, since it soon becomes infeasible to repeatedly find a globally optimal solution based on the ever-growing volume of data acquired from a live camera. Some real-time methods which can be classed as visual odometry (e.g. [19], [21]) choose to 'forget' information from past measurements beyond a sliding time window. The result is highly accurate local motion estimation due to the ability to cope with a large number of feature measurements per frame, but drift over extended sequences. This problem has recently been successfully mitigated by the use of 'keyframes': a subset of representative images and camera poses selected from the continuous stream and subject to global optimisation with the rest of the trajectory related to these [16], [17].

Alternative real-time methods for visual mapping based on sequential probabilistic filtering (e.g. [7], [11]) aim to 'summarise' the information gained from past images with a probabilistic state. This uncertain estimate of the camera and map state can be combined with the information from each new image in a weighted average of fixed complexity at each time-step. It turns out however that the accurate probabilistic representation of uncertainty which is required here is computationally expensive in a way which scales poorly with the number of mapped features. For this reason methods such as [7] only map features relatively sparsely. The most successful solution to this problem has been, as in real-time SLAM research using other sensors (e.g. [1], [2], [4], [15], [24]), to split a large map into several conditionally independent visual submaps which can be processed separately (e.g. [6], [12]). Perhaps the most successful approach has been Eade and Drummond's sophisticated real-time monocular SLAM system [12], [13] which connects submaps (here called 'nodes') with a higher level graph structure estimating their relative locations.

So keyframes or submaps are sparsifying approximations which permit real-time implementation of globally consistent mapping. But in the case of either keyframes or submaps, there remains the question of how to choose their locations and scope.

B. The Special Character of Visual Maps

A little consideration makes it clear that visual sensing is not in general conducive to a straightforward division of a scene into block-like submaps for the purposes of efficient map processing, as has proven successful with other sensors. Laser range-finders and sonar sensors have strictly limited ranges of measurement, and setting submap sizes which relate closely to this is a sensible strategy — features located farther apart than this range will not be simultaneously observed. There are other potential heuristic strategies for the choice of submap boundaries: an upper bound on the number of features, or bounded uncertainty (or deviation from linearity as in [12]) within a submap. In keyframe approaches, all of the scene elements visible from a particular camera pose are implicitly grouped together for the purposes of estimation, independent of their distance from the camera. This can cope with a range of depths, but is still a somewhat arbitrary grouping.

Consider for instance the example of a large, cluttered room (perhaps a cafeteria), browsed and mapped by a mobile camera carried by a robot or person. The camera will view parts of the room from different distances to obtain different levels of detail: a table may be framed from close-up, or the camera may move even closer to inspect particular objects. The periphery of these views though may simultaneously be filled with distant walls or even the outside scene beyond the windows. Different features in a scene tend to have more strongly correlated estimates in a map when they are regularly co-observable by the moving sensor, but this is not always the case if they give different information about camera location. Similarly, features in almost the same scene location but measured from different camera positions may be uncorrelated.

C. Determining Hierarchical Map Structure

In this paper we propose a straightforward and absolute measure for the level of correlation between features in a mapping scenario based on the mutual information of predicted measurements. We show that this automatic inference of structure can in fact easily go beyond a single level of submaps to deduce a full hierarchy of correlation relations via a tree decomposition. In fact, many of the most exciting recent approximate but super-efficient SLAM algorithms [14], [22], [23] are tree-like in nature, showing the additional power this gives.

The tree structure encodes a hierarchy of correlation levels between features which permits their grouping into sets with a user or application-settable coarseness or fineness, from one extreme where all features are considered as independent and unrelated to the other where they will all be grouped together. In between, features will be accumulated into clusters which gradually join into a single whole.

It is important to note that the hierarchical structure which our method discovers is that of the *probabilistic map*, not the a fundamental property of the scene itself. The structure depends on the motion of the camera, priors which we have about how the camera moves, and its imaging properties such as resolution and field of view. In a map built using an omnidirectional camera, for instance, we might expect simultaneously observable features on opposite sides of a robot to be regularly highly correlated in measurements and that they would be clustered together, while in a map built using a camera with a narrow field of view they would be distant in the tree. We consider that this dependence on the specifics of the camera and motion is a strength of the approach, not a weakness.

II. Measurement Prediction and Mutual Information

The basis for our analysis of visual map structure is the pairwise mutual information (MI) between candidate image feature measurements at each frame of a filtering sequence. Mutual information can be understood as an absolute, normalised measure of degree of correlation. Strictly, the MI of two uncertain variables is the number of bits of information expected to be gained about one of them upon determining the precise value of the other.

Following the formulation of [8], we consider making image measurements of a scene of which the current state of knowledge is modelled by a probability distribution over a finite vector $\mathbf{x}$ stacking camera and map parameters. In an image, we are able to observe measurable projections of the scene state which we call *features*. A measurement of feature i yields the vector of parameters $\mathbf{z}_i$; for instance the two-dimensional image location of a point feature. A likelihood function $p(\mathbf{z}_i|\mathbf{x})$ models the measurement process for a particular camera and feature type.

When a new image arrives, we can project the current probability distribution over the state parameters $\mathbf{x}$ into feature space to *predict* the values and distributions of all the possible feature measurements which can be made. In SLAM, this current distribution will be the result of the application of a motion model representing temporal dynamics or odometry sensing to the distribution resulting from the previous frame. By building the stacked vector $\mathbf{z}_T = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots \end{pmatrix}^\top$ containing all candidate feature measurements, we can calculate the joint density over all measurement predictions:

$$p(\mathbf{z}_T) = \int p(\mathbf{z}_T|\mathbf{x})p(\mathbf{x})d\mathbf{x} \ . \tag{1}$$

This joint prediction can be used for probabilistic data association (matching), in either batch [20] or sequential [3] forms. These algorithms benefit from the fact that the level of *correlation* between the predictions of different feature locations is high typically, since they all depend on common parts of the scene state $\mathbf{x}$ — most significantly, uncertainty in camera location induced by the motion model. The pairwise mutual information between the predicted feature locations is a normalised measure of these correlations.

Following the notation of Mackay [18], the (MI) of variables $\mathbf{z}_i$ and $\mathbf{z}_j$ is:

$$\begin{aligned} I(\mathbf{z}_i; \mathbf{z}_j) &= E\left[\log_2 \frac{p(\mathbf{z}_i|\mathbf{z}_j)}{p(\mathbf{z}_i)}\right] & (2) \\ &= \int_{\mathbf{z}_i, \mathbf{z}_j} p(\mathbf{z}_i, \mathbf{z}_j) \log_2 \frac{p(\mathbf{z}_i|\mathbf{z}_j)}{p(\mathbf{z}_i)} d\mathbf{z}_i d\mathbf{z}_j \ . & (3) \end{aligned}$$

By this general definition, the MI between two candidate measurements can be calculated whatever the functional forms of $p(\mathbf{x})$ and $p(\mathbf{z}_i|\mathbf{x})$.

We can now define the Mutual Information (MI) matrix as in (4), so that every off-diagonal entry is calculated based on the covariance relating the predicted measurements in $\mathbf{z}_T$. The MI matrix represents the expected information gain of a candidate measurement given the exact state of another. If N is the total number of candidates then:

$$\mathtt{I}(\mathbf{z}) = \begin{bmatrix} * & I(\mathbf{z}_1; \mathbf{z}_2) & \dots & I(\mathbf{z}_1; \mathbf{z}_N) \\ I(\mathbf{z}_2; \mathbf{z}_1) & * & \dots & I(\mathbf{z}_2; \mathbf{z}_N) \\ \vdots & \vdots & \vdots & \vdots \\ I(\mathbf{z}_N; \mathbf{z}_1) & I(\mathbf{z}_N; \mathbf{z}_2) & \dots & * \end{bmatrix} . \quad (4)$$

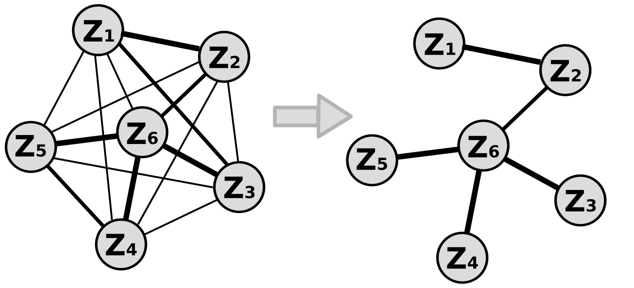
This matrix is the basis for all of the analysis we will conduct later to determine map structure. The matrix is symmetric and the elements on the diagonal are meaningless, and therefore filled with $*$'s.

The MI matrix has values for every pair of observed features on each frame, and we can use it to analyse correlations on a frame by frame basis. When tracking a sequence, we can build up a model of measurement correlations between all features in the map by accumulating average mutual information scores for each combination of two features in a large MI average matrix. Note here that any features which are never co-observed will have a mutual information score of zero in this matrix. Features that have been covisible throughout a substantial number of frames and have been moving consistently share high mutual information links, whereas features with significant depth difference in the scene will have a corresponding parallax difference in image space and therefore weaker mutual information.

An important point about this information-theoretic measure of correlation is that it gives a valid, absolute value for arbitrary combinations of features of any type. In [8] an example is given where the mutual information for a joint set of edge and point features is used to sensible guide active search.

A. Gaussian Case

While the MI formulation is valid for any type of distribution, in this section we derive the specific form for the case where the PDFs describing knowledge of $\mathbf{x}$ and $\mathbf{z}_T$ can be approximated always by single multi-variate Gaussian distributions. The measurement process is modelled by $\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \mathbf{n}_m$, where $\mathbf{h}_i(\mathbf{x})$ describes the functional relationship between the expected measurement and the scene state as far as understood via the camera measurement model, and $\mathbf{n}_m$ is a Gaussian-distributed vector representing unmodelled effects (noise) with covariance $\mathtt{R}_i$ which is independent for each measurement. The vector $\mathbf{z}_T$ which stacks the predicted measurements can be calculated along with its full covariance $\mathtt{S}$, usually known in tracking parlance as the innovation covariance matrix:



(a) Complete MI graph (b) Chow-Liu tree approximation

Fig. 1. The approximation of a joint PDF $p(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_6)$ by second order conditionals and marginal distributions yields a tree. Here, (a) shows the complete MI graph where thicker links represent higher mutual information between the nodes they connect, and (b) is the optimal such approximation maximising the total information preserved in the tree as suggested by Chow and Liu.

$$\mathbf{z}_T = \begin{pmatrix} \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{h}_1(\hat{\mathbf{x}}) \\ \mathbf{h}_2(\hat{\mathbf{x}}) \\ \vdots \end{pmatrix} \quad (5)$$

$$\mathtt{S} = \begin{bmatrix} \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top + \mathtt{R}_1 & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top & \dots \\ \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top + \mathtt{R}_2 & \dots \\ \vdots & \vdots & \end{bmatrix} \quad (6)$$

The correlations between different feature predictions mean that generally $\mathtt{S}$ will not be block-diagonal but contain off-diagonal correlations between the predicted measurements of different features.

With this single Gaussian formulation, the mutual information in bits between any two predicted measurements $\mathbf{z}_i$ and $\mathbf{z}_j$ can be calculated according to this formula ([8]):

$$I(\mathbf{z}_i; \mathbf{z}_j) = \frac{1}{2} \log_2 \frac{|\mathtt{P}_{\mathbf{z}_i \mathbf{z}_i}|}{|\mathtt{P}_{\mathbf{z}_i \mathbf{z}_i} - \mathtt{P}_{\mathbf{z}_i \mathbf{z}_j} \mathtt{P}_{\mathbf{z}_j \mathbf{z}_j}^{-1} \mathtt{P}_{\mathbf{z}_j \mathbf{z}_i}|} \ , \quad (7)$$

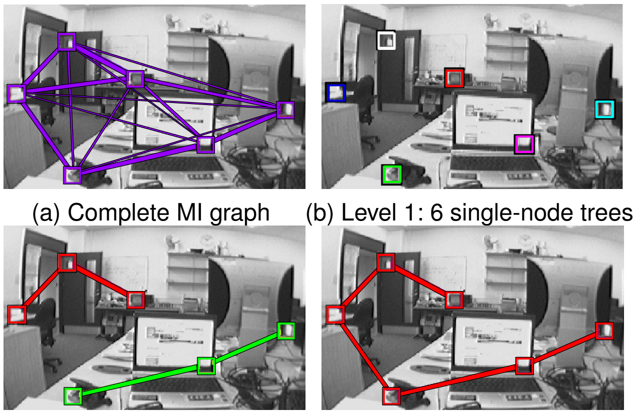
where $\mathtt{P}_{\mathbf{z}_i \mathbf{z}_i}$, $\mathtt{P}_{\mathbf{z}_i \mathbf{z}_j}$, $\mathtt{P}_{\mathbf{z}_j \mathbf{z}_j}$ and $\mathtt{P}_{\mathbf{z}_j \mathbf{z}_i}$ are sub-blocks of $\mathtt{S}$. This representation however can be computationally expensive as it involves matrix inversion and multiplication so we simplify the formulation that was used in [3] which exploits the properties of mutual information:

$$\begin{aligned} I(\mathbf{z}_i; \mathbf{z}_j) &= H(\mathbf{z}_i) - H(\mathbf{z}_i|\mathbf{z}_j) = H(\mathbf{z}_i) + H(\mathbf{z}_j) - H(\mathbf{z}_i, \mathbf{z}_j) & (8) \\ &= \frac{1}{2} \log_2 \frac{|\mathtt{P}_{\mathbf{z}_i \mathbf{z}_i}||\mathtt{P}_{\mathbf{z}_j \mathbf{z}_j}|}{|\mathtt{P}_{\mathbf{z}_i \mathbf{z}_i} \mathtt{P}_{\mathbf{z}_j \mathbf{z}_j} - \mathtt{P}_{\mathbf{z}_i \mathbf{z}_j} \mathtt{P}_{\mathbf{z}_j \mathbf{z}_i}|} \ . & (9) \end{aligned}$$

III. Tree Factorisation

A probabilistic estimate of the values of a set of variables $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}$ given background information I is most generally specified by a joint density function over all of those variables:

$$p(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N) = f(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N) \ . \quad (10)$$


(a) Complete MI graph

(b) Level 1: 6 single-node trees

(c) Level 2: 2 trees

(d) Level 3: Chow-Liu tree

Fig. 2. Building the Chow Liu tree in a real scene. In (a) is the graphical representation of the Mutual Information matrix for this frame. Every link represents presence of mutual information between the features it connects, with thickness corresponding to magnitude. The process of building the Chow Liu tree as an approximation to this graph is bottom-up, so at level 1 in (b) we have 6 different trees each comprising of a single feature. In level 2 in (c) each tree is grouped with the tree that is most information dependent. Finally in (d) all features lie in one tree, the Chow Liu tree.

One possible approximation to a general joint probability density is the factorised form in (12).

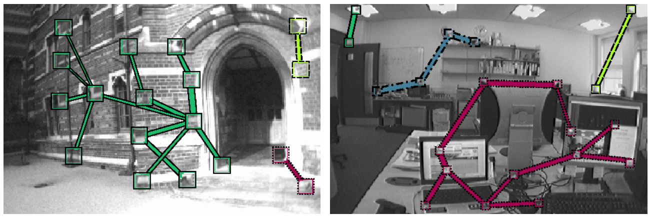
$$p(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N) = p(\mathbf{z}_N) \prod_{i=1}^{N-1} p(\mathbf{z}_i | \mathbf{z}_{i+1} \ldots \mathbf{z}_N) \quad (11)$$

$$\approx p(\mathbf{z}_N) \prod_{i=1}^{N-1} p(\mathbf{z}_i | \mathbf{z}_{i+1}) \quad (12)$$

Figure 1 shows that this approximation can be interpreted as a tree-shaped model of probabilistic links between variables (each link representing a conditional density function of just the two connected variables). Chow and Liu showed in their 1968 paper [5] that a full, joint probability density of a set of variables can be optimally approximated as a product of second-order conditionals and marginal distributions (as in (12)) chosen to minimise the difference in Kullback-Leiber divergence. The first-order dependency tree yielding from this selection of links between variables is equivalent to the maximum spanning tree[1] of the Mutual Information graph (the weights in this graph are defined by the elements of the matrix in (4)).

IV. Inferring Hierarchical Structure from the Tree

Have deduced undirected tree structure linking the features, there remains the question of how to use this to infer hierarchical clusters. One option would be to somehow choose a root feature and then 'hang' the tree from this. By choosing a number of levels down from this root we could fix where to lop off branches, all nodes further down each branch forming a cluster. Alternatively, we could use a threshold on the MI scores on branches of the tree, cutting all those weaker than a certain value to divide the tree into clusters. In experimentation, while this approach has some nice properties it tends to leave many features alone in clusters of one.


(a) Planar scene

(b) Scene with different depths

Fig. 3. Clustering features based on data obtained from a single frame. In (a) is a typical frame of the Keble College Sequence, tracking features on the wall. Since this is a planar scene, the distribution of mutual information of features with each other is uniform, therefore the clustering result is based on image proximity. On the other hand in (b) is a scene with more interesting structure hence the clusters are also based on depth of features.

Instead, here we propose a simple bottom-up procedure where features are progressively grouped in a manner similar Chow and Liu's original algorithm to build the spanning tree. The goal being to identify image regions of high mutual information density, we consider an example where N features have been tracked in a sequence of frames and start join features together. We start off as if these features were completely uncorrelated. All off-diagonal entries in the MI matrix would then be zero and therefore at this stage we have N different trees each comprising of a single feature. Jumping a level up the hierarchy, the aim is to link each tree to the tree with which it is sharing the strongest tie so that no cycles are introduced. Repeating this process, we reach the root of the hierarchy where all features lie in the same tree, the Chow Liu tree. Figure 2 shows a real, simplified example of the step-by-step building process of this tree.

Following the analysis to infer the scene structure in a single frame, we can expand this idea to a sequence of frames. Keeping a running average of the mutual information links between features in the map, we accumulate information on features that were coobserved at any instant. We can then build the Chow Liu tree over the whole map to automatically discover areas of high mutual information density in a hierarchical manner. It is worth noting that at any frame we only need to calculate the MI matrix of the measurable features in that frame, therefore the cost is tractable, since all data needed is evaluated in image space.

V. Results

We demonstrate our algorithm on several different visual maps generated from a hand-held camera using a standard configuration of the freely available MonoSLAM system for real-time single camera SLAM [7], [9]. MonoSLAM uses the Extended Kalman Filter (EKF) to incrementally construct a probabilistic map of visual point features represented by a single joint Gaussian distribution. At each new frame MonoSLAM selects features for measurement based

[1] The acyclic path connecting all nodes in a weighted graph which yields the maximal sum of weights.

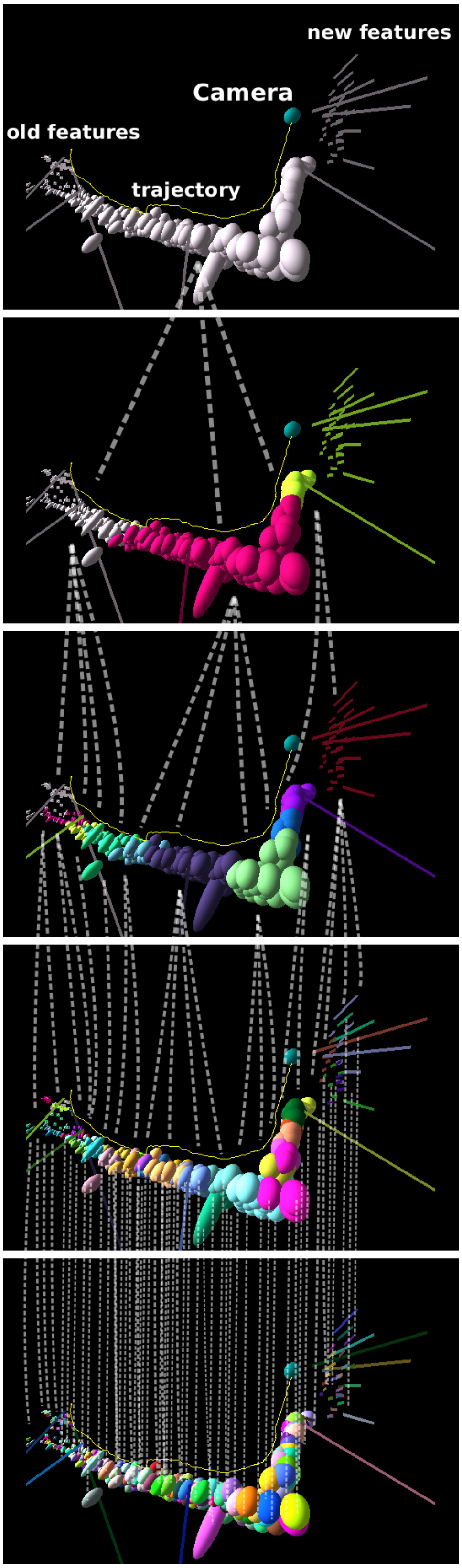


Fig. 4. Discovering the map structure of an exploratory sequence taken in Liddon Quad of Keble College. At the top of the root of the tree (top image) is a single map containing all 329 features tracked during the 2500 frame sequence. Moving down the hierarchy each map splits into several submaps until each feature lies in a different submap (bottom image).

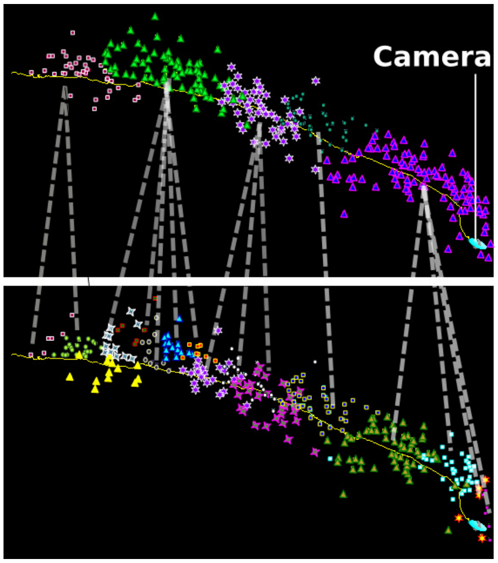


Fig. 5. Submapping in a corridor sequence tracked with a forward-looking camera. Here are two intermediate levels of the tree hierarchy where features are visible in both sides of the camera's trajectory, therefore left and right hand side features are grouped into the same submap. The submaps formed here have overlaps due to the covisibility of features belonging to different submaps during tracking, in contrast to the Keble sequence in figure 4 where submaps are more discrete since the camera is looking sideways. Note that the feature uncertainties are not displayed here for the sake of clarity.

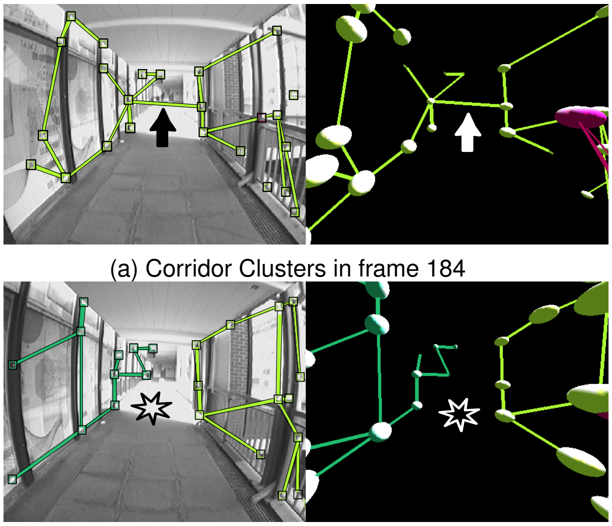
on whether they are predicted to lie within the camera's field of view and whether the camera is predicted to be within a set of bounds for each feature on motion (inducing scale changes and warping) where correlation matching is expected to be possible. The innovation covariance matrix $\mathtt{S}$ is calculated on every frame of MonoSLAM as part of the active feature matching (data association) process so there is little additional computational cost incurred by our tree construction algorithm.

We present analysis of several single frames and extended sequences at 30Hz which draw attention to the behaviour of the algorithm and indicate the valuable role it can play in automatic submap definition. Finally, we compare our method to naive submapping through a quantitative analysis. The video accompanying this paper, illustrates the run-time clustering in selected parts of the sequences discussed in this section.

A. *Single Frame Analysis*

As a proof of the concept of this paper, we performed the simplest application of our tree-based clustering; a Chow-Liu tree is built using data from a single frame only.

Once the correlations between features have been settled and the map has converged, then so are the mutual information links between them. Therefore, building the Chow-Liu tree we can infer clusters that are conceptually consistent. Image 3 is a demonstration of grouping features based on their proximity in image space in the case of a planar scene and the distinction of background/foreground features in a scene with significant depth positions.



(a) Corridor Clusters in frame 184

(b) Corridor Clusters separate in frame 208

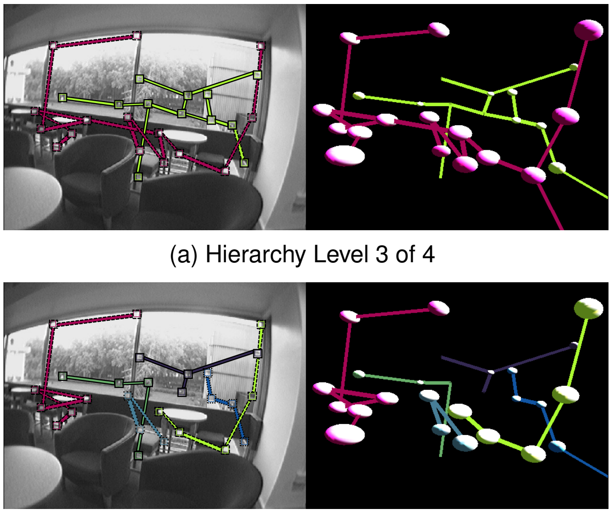
Fig. 6. Typical clusters in the corridor sequence. In (a) there are 2 clusters present (bright green and red). The arrow points to a link that breaks in (b) after 24 frames, since the features it used to connect no longer seem to be moving in the same way as before. As a result a new (darker green) cluster is formed to distinguish the features on the left wall from the ones on the right. Tracking in a corridor-like sequence with a forward-pointing camera, means that the distant features will appear near the centre of the image and grouped together with features they appear to be close by in image space. As the camera moves further such features gradually move away from each other changing the distribution of mutual information links in the Chow Liu tree, therefore changing the structure of clusters.

However, the interest here is to infer meaningful and consistent submaps through time, where features are constantly added, deleted and updated in the map, hence follows the analysis on sequences of frames.

B. Sequence Analysis

1) Sideways Exploration: Here we analyse a segment of the image sequence of Clemente *et al.* [6] taken by a hand-held camera moving sideways around a large college quadrangle, moving at a steady walking speed constant speed while observing a wall at approximately constant depth. We call the sequence exploratory because the camera moves progressively and does not return to previously visited positions in the segment. This sequence is of interest because its simple nature makes the 'ideal' map structure a clear case of approximately regular metric division, as implemented explicitly in [6] by bounding the number of features in each submap at a fixed value.

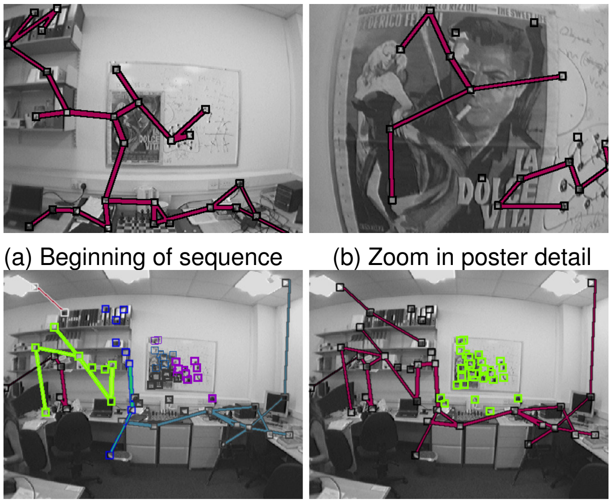
Figure 4 shows the grouping of features at all levels of the Chow-Liu tree formation; each feature belongs to a different submap at the leaves of the hierarchical tree, and then they gradually team-up to form a single submap. Due to the roughly constant speed of the camera and the regular presence of features on the observed wall, the distribution of mutual information links is uniform and therefore the clusters forming in the intermediate levels of the hierarchy are fairly similar in size.



(a) Hierarchy Level 3 of 4

(b) Hierarchy Level 2 of 4

Fig. 7. Clustering using accumulated mutual information of features over a sequence of frames. Both (a) and (b) show the same scene of tracking features with significant difference in depth estimates. Images on the right show the 3-D view of the map and on the left are the corresponding camera-views with positions and uncertainties of features projected in image space. The links displayed are the segments of the Chow-Liu tree present at the corresponding level of hierarchy. In (a) the two clusters demonstrate the background-foreground separation as suggested by the feature uncertainties (far features have much smaller uncertainty than closer ones as they are only expected to move by a small amount in image space from one frame to the next). Moving a step deeper in the hierarchy, these clusters split into smaller ones in (b) showing regions of higher mutual dependency.



(a) Beginning of sequence

(b) Zoom in poster detail

(c) End: cluster detail

(d) End: main clusters

Fig. 8. Maintaining the scene's detail in the tree structure. At the beginning of the sequence in (a), all measurable features appear to lie on a plane, hence are grouped in a single tree. In (b) the camera zooms in the poster and more features are initialised to track the close-up view. In (c) and (d) is the final tree structure at the end of the sequence. In (d) it is evident that all detail-features of the poster have been clustered in a distinct group from the rest of the features, and (c) shows more detail one level down the tree hierarchy.

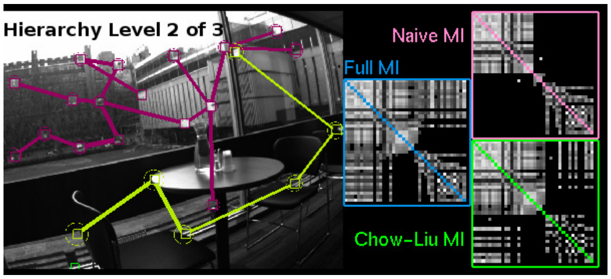


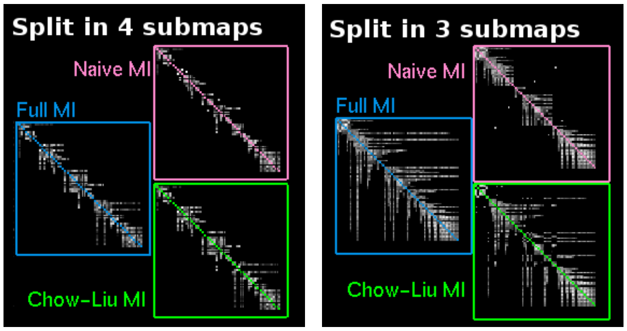
Fig. 9. Comparing the quality of our clustering method with naive submapping. On the left is the camera view with the features tracked in this frame (colour depicts cluster membership) and on the right are the matrices of pairwise mutual information links between all features in the map built so far. Brighter pixels denote stronger mutual information links in measurement space. The true matrix of all such links is displayed in the blue box as the ‘Full MI’. The other two matrices display approximations to the Full MI according to the submapping scheme used. It is evident that our clustering method preserves far more structure in the distribution of MI links rather than the naive approach due to the careful selection of the cluster partition. Here, splitting the map in two clusters with the naive approach we capture 55% of all the links of the Full MI whereas using our Chow-Liu tree based method we capture 81%.

2) Forward Exploration: The next example is an exploratory sequence from a forward-moving hand-held camera. The additional interest here is in the presence of features close to the centre of expansion in the middle of the image which are very distant and therefore remain visible for long periods of time while the main quantity of features towards the edges of the image quickly pass out of the field of view.

Figure 6 shows an example where distant features near the centre of the image appear close by in image space, therefore belong to the same tree but as the camera moves closer, the link between them breaks to form two different clusters on each side of the wall. Figure 5 is a full map view of all the features tracked along the corridor and the clusters formed. The difference with the Keble sequence map presented above is that features appear in both sides of the trajectory here as the camera is facing forward, and also, there is overlap between clusters in state space due to their covisibility in image space.

3) Loopy Browsing of a Scene with Various Depths: Here is an example of our clustering method applied on a scene with a substantial disparity in depth. Figure 7 shows a typical frame of the sequence co-viewing close-by and distant features through a cafeteria’s window. The features on the window frame appear close to the features outside the window in image space, but the correlation between these two groups is weakened over time due to their difference in parallax thus are clustered in separate submaps.

4) Level of Detail in a Natural Scene: In this experiment the camera is moved forward from a position in the middle of a room to closely inspect a poster and wall first seen from the distance. As the camera approaches, features are mapped increasingly densely on the surface of the poster. Meanwhile, features initialised while the camera was distant from the wall become unmeasurable and are deselected by MonoSLAM.



(a) Sideways exploration (b) Forward exploration

Fig. 10. The distribution of pairwise MI links before and after submapping using either naive or our Chow-Liu tree based clustering methods. In the sideways exploration sequence the camera is constantly exploring new areas therefore the block-diagonal structure of the MI links in (a) whereas in (b) features initialised early in the image remain visible for a long time building correlations with other features seen along the camera’s path.

C. Quantitative Analysis

As a means of demonstrating the effect of selecting the cluster partitions carefully, we superimpose naive submapping with our Chow-Liu tree based clustering method at different levels of the clustering hierarchy when tracking for 1000 frames in each of the sequences 1 to 3 of subsection V-B. At the end of each sequence, we record the effect of partitioning the map into an equal number of submaps in both clustering schemes (naive submapping splits the map into regular-sized clusters of features, in the order that they were initialised into the system). Each scheme provides an approximation of the distribution of the pairwise MI links between all features, therefore as a comparison measure we use the ratio of the sums of the MI links preserved over all the MI links present in the whole map. The table below, shows these ratios as percentages at each level of the hierarchy built using our clustering approach.

		Pairwise measurement MI captured	
HIERARCHY	SUBMAPS	NAIVE APPROX	CHOW-LIU APPROX
Scene with various depths			
4 of 4	1	100 %	100 %
3 of 4	2	50.78 %	84.50 %
2 of 4	7	21.43 %	35.67 %
1 of 4	38	0 %	0 %
Sideways exploration			
4 of 4	1	100 %	100 %
3 of 4	4	74.70 %	93.83 %
2 of 4	10	51.67 %	75.38 %
1 of 4	60	0 %	0 %
Forward exploration			
4 of 4	1	100 %	100 %
3 of 4	3	74.51 %	85.47 %
2 of 4	18	33 %	43.70 %
1 of 4	111	0 %	0 %

For all three sequences, the highest hierarchy level corresponds to the whole map, therefore there is no approximation at all whereas in the lowest hierarchy level every submap contains a single feature preserving no links between them.

The results demonstrate that in all cases, selecting the submap partitions carefully pays off in terms of achieving a better approximation to the full map. The biggest difference between the two submapping schemes is recorded when splitting in two the map built for the Scene with various depths where submapping with the Naive approach the preserved MI links sum up to 51% of the total MI present in the whole map, whereas our approach preserves 85% of the initial distribution. Figure 9 shows an example frame of the sequence along with a visual representation of the matrix of MI links before and after each approximation. The mostly exploratory nature of the other two sequences results in a sparser distribution of links in the map as demonstrated in Figure 10 therefore the loss of both approximation is smaller.

VI. Conclusion

We have shown that straightforward calculation of the mutual information of feature measurements, temporal averaging and tree construction provide a computationally efficient way to automatically extract the full hierarchical correlation structure of a visual map as it is built.

Our experiments show that the resulting hierarchical structure displays characteristics which agree with the expected behaviour in 'obvious' cases such as simple exploration where regular division is appropriate, but which also capture much more subtle effects in scenes and camera motions with large ranges of depth or level of detail. Future work involves developing a filter based on this submapping approach for efficient SLAM. Also we will consider exploiting appearance information along with geometry to refine the definition of submaps as a means of perhaps understanding the semantic nature of each submap.

Acknowledgements

This research was supported by EPSRC grant GR/T24685/01. We are grateful to Ian Reid, José María Montiel, José Neira, Javier Civera and Paul Newman for very useful discussions.

References

[1] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[2] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only SLAM in large unstructured environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[3] M. Chli and A. J. Davison. Active matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[4] K. S. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal of Robotics Research (IJRR)*, 18(2):3–19, January 1999.

[5] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[6] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós. Mapping large loops with a single hand-held camera. In *Proceedings of Robotics: Science and Systems (RSS)*, 2007.

[7] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2003.

[8] A. J. Davison. Active search for real-time vision. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005.

[9] A. J. Davison, N. D. Molton, I. D. Reid, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052–1067, 2007.

[10] F. Dellaert. Square root SAM. In *Proceedings of Robotics: Science and Systems (RSS)*, Cambridge, USA, June 2005.

[11] E. Eade and T. Drummond. Scalable monocular SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[12] E. Eade and T. Drummond. Monocular SLAM as a graph of coalesced observations. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007.

[13] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular slam. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2008.

[14] U. Frese. Treemap: An $o(logn)$ algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103–122, 2006.

[15] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. Accepted for publication in IEEE Transactions on Robotics (T-RO), 2008.

[16] G. Klein and D. W. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[17] K. Konolige and M. Agrawal. FrameSLAM: from bundle adjustment to real-time visual mapping. Accepted for publication in IEEE Transactions on Robotics (T-RO), 2008.

[18] D. Mackay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[19] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real-time localization and 3D reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[20] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics and Automation*, 17(6):890–897, 2001.

[21] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[22] M. A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1157–1164, 2003.

[23] L. M. Paz, P. Jensfelt, J. D. Tardós, and J. Neira. EKF SLAM updates in o(n) with divide and conquer SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[24] J. D. Tardós, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research (IJRR)*, 21(4):311–330, April 2002.

[25] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Cambridge: MIT Press, 2005.

[26] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.