Using Scene Similarity for Place Labelling

I. Posner, D. Schroeter, and P. Newman

Robotics Research Group, University of Oxford, {hip, ds, pnewman}@robots.ox.ac.uk

Abstract. This paper is about labelling regions of a mobile robot's workspace using scene appearance similarity. We do this by operating on a single matrix which expresses the pairwise similarity between all captured scenes. We describe and motivate a sequence of algorithms which, in conjunction with spatial constraints provided by the continuous motion of the vehicle, produce meaningful workspace segmentations. We provide detailed experimental results from various outdoor trials.

1 Introduction and Related Work

We would like a mobile robot to group and label similar regions of its workspace. As it traverses through an extended workspace it is likely to pass through regions with a distinct nature and character. Classifying these regions into natural partitions not only adds higher order meaning to any maps built by the vehicle but can also constrain the computation required in localisation. We show how combining topological constraints with scene similarity can lead to a useful and credible clustering of scenes into distinct classes.

The bulk of recent research into autonomous platform navigation has used sensors to extract and eventually infer solely metric information. There is no doubt that great progress has been made, particularly in the context of the SLAM problem. However, appearance-based techniques may also have an important role to play [1–3] and in this paper we discuss how appearance-based reasoning can be used for workspace classification. We use the term “scene” to mean the local workspace of the vehicle as captured by onboard sensors (which could be laser, cameras, radar or sonar). By defining a suitable metric, “scene similarity” becomes a scalar ranging between zero and unity representing utter dissimilarity through to carbon-copy replication. Finding representative classes of scenes based on such a measure is a particular instance of a problem commonly referred to as Knowledge Discovery in Databases (KDD):

> *”...the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. KDD encompasses a number of different technical approaches, such as clustering, data summarizing, learning classification rules, finding dependency networks, analyzing changes, and detecting anomalies.” Piatetsky-Shapiro et al. [4, page 77]*

The related literature is extensive indeed but document retrieval [5], indexing [6,7] and appearance-based image classification [8–10] are most relevant to the work presented here. Commonly supervised learning methods are applied, where models are trained using labelled data, and evaluated on a separate set of test data [11]. Our goal however is to enable a mobile robot automatically to extract meaningful concepts from the given data, and we investigate to what extent unsupervised learning can help to solve this problem. In particular, we seek to find algorithms that produce consistent representations for distinctive places or common environment classes. In this sense, our work differs from document retrieval or indexing where queries are answered by finding the most similar documents. We note that the representation of underlying themes is not explicitly generated, but also seldom necessary.

There exist a great deal of literature on unsupervised and semisupervised learning Though it is beyond the scope of this paper to give an exhaustive review, scene and object classification in computer vision has made great progress in the last decade [9,10,12]. For example, Bosch et al. [8] have successfully used *Probabilistic Latent Semantic Analysis (pLSA)* [7] to automatically learn models of objects and environment classes. Due to its computational complexity, methods using EM learning to fit model parameters are not particularly well suited for mobile robotic applications. Since the robot moves about the environment and gathers the data in a sequential manner, there is a temporal ordering in the data that implies local spatial relationships. To the best of our knowledge, this topological information has not been exploited within the robotics research community.

In the following sections we will describe and provide experimental results of a system for semantic labelling of contiguous workspace locations. Our method is designed for use on mobile platforms, being swift and able to explicitly take advantage of the vehicle motion. Its input is the pairwise similarity between all scenes and we shall emphasise the consequences of when the fidelity of this measure is imperfect. While we detail the use of images, the method is equally applicable for any sensor modality for which an appearance-based similarity measure can be defined.

2 From Scenes to Similarity

Consider the impressions of two scenes, S_u and S_v; we do not at this stage need to describe in detail what constitutes these impressions – they could for example be images, laser scans or radar sweeps. However, we shall often use the case of an image as a concrete example of S_u to illustrate our adopted approach. From each scene, S_u, n regions of interest are extracted and each region encoded by a suitably chosen descriptor vector. This mapping $\mathcal{D} : S_u \rightarrow [d_1 \cdots d_n]$ transforms the scene into a list of vectors whose length n is a function of the complexity of the particular scene in question. In the case of an image, for example, we use a Harris Affine ROI detector [13], because of its wide baseline invariance, and a SIFT descriptor [14] yielding a set of 128-dimensional vectors. As in [10], the next step involves the clustering of all descriptor vectors from a training set of input data. This operation is an off line task and yields a set of cluster centres $\mathcal{V} = [\hat{d}_1, \hat{d}_2, \cdots \hat{d}_{|\mathcal{V}|}]$, which collectively are often referred to as a "code-book" ,"visual vocabulary" or "bag-of-words". The size of the vocabulary is $|\mathcal{V}|$. Words that are ubiquitous, perhaps appearing in every scene, have reduced descriptive power compared to those that occur rarely. When we come shortly to using the word-content of scenes to measure inter-scene similarity, common words should have less weight. A commonly used weight scheme is the Inverse Document Frequency (IDF) [15]. Here each word $\hat{d}_i$ is assigned a weight $w_i = log\frac{N}{n_i}$ where N is the total number of scenes (images) gathered and n_i is the number of scenes in which $\hat{d}_i$ occurs. We can now define a similarity function $0 < \mathcal{S}(u, v) < 1$ between two scenes S_u and S_v. Each scene is quantised into a vector of length $\mid \mathcal{V} \mid$ where the i^{th} element is w_i if word $\hat{d}_i$ is present and zero otherwise. A simple choice for $\mathcal{S}$ is then the cosine distance between the two vectors — scenes that have no common features will have zero similarity and those with complete intersection will have a similarity of one. We emphasise that what we have described here is just one of many ways to formulate $\mathcal{S}$. We do not however mean to be prescriptive; in general we require only that some function $\mathcal{S}$ exists which expresses the similarity.

A construct central to our work is the similarity matrix M. Each element, $\mathtt{M}_{i,j}$, is the similarity between scenes i and j. Figure 1 shows typical similarity matrices constructed using visual and laser scenes. The dominant diagonals are caused by all scenes being self similar. The off-diagonal stripes, where visible, are indicative of loop closures and the banding is due to broad, wideband, scene similarity.

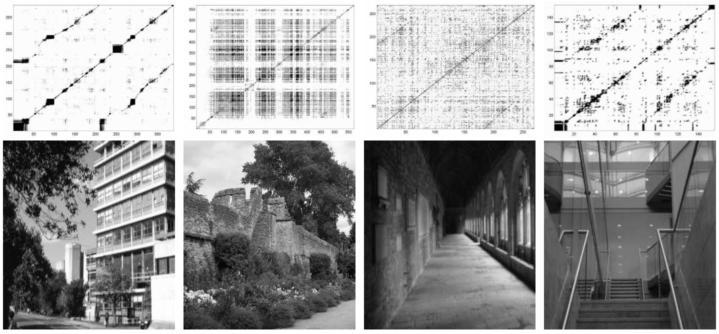

Fig. 1. Four similarity matrices for four markedly different workspaces. From left to right (with sensor used) : the exterior of a tower block (camera), an excursion around some formal gardens (camera), a loop around architecturally uniform cloisters (camera), and a loop around the interior of a building (2D laser).

3 Scene Classification

The annotation of maps with semantic information carries intrinsic value and is being researched from a variety of perspectives. See, for example, [16] which uses boosting and spatial smoothing to classify indoor workspaces. As an additional motivation, learning of outdoor scene classes presents the opportunity to reduce the computational expense involved in loop-closure detection by partitioning the search space. If, for example, recent images appear to belong to scene class *Park* then all images of scene type *Building* can be discounted when looking for loop closure — park-like scenes rarely suggest loop closures within buildings.

The use of just the similarity matrix to attempt place labelling is mainly motivated by the following two reasons. Firstly, in our SLAM system, this matrix has already been built for use in loop closure detection as described in [2]. Secondly, we are working towards replacing the similarity entries in M with probabilities — which would allow principled fusion of scene appearances as perceived by a heterogenous sensor suite. Nevertheless, the central data structure will remain a matrix encoding inter-scene equivalences. This representation, while compact and advantageous for some tasks, is not sufficient for some of the more common clustering algorithms (such as k-means). Although multidimensional scaling [17] can be employed to transform the data into a metric space, this transformation comes at the cost of inaccurately representing the relative positions of individual exemplars and consequently results in unsatisfactory clustering performance.

Interpreting the similarity matrix as a graph G where the vertices are image locations and the edges are the similarities between images, *Normalised Graph Cut (NGC)* [18] can be applied recursively, terminating when no clear partitioning exists.[1] Aside from the cost of solving multiple eigen-problems we found the procedure to suffer from two issues. One difficulty was the convergence and stability. Secondly, it was not always clear how to decide if a putative partition was valid.

[1] NGC finds a bipartite partition of G which minimises a metric that considers both the total dissimilarity between different groups and the total similarity within the groups.

This became increasingly problematic as the depth of the recursion increased and accordingly the size of the (sub)-graph being processed decreased.

Hierarchical clustering [19–21] is a well established technique based directly on a distance metric. Commonly, grouping of N data is achieved by iteratively merging clusters starting with N individuals (*agglomerative*). At each stage individual clusters are fused which are in some sense most 'similar'. The choice of similarity measure has important implications on clustering performance. *Single linkage* considers the maximum inter-cluster pairwise similarity and generates a minimum spanning tree over the data. Single linkage does not take account of cluster structure and tends to produce unbalanced and elongated clusters (chaining effect). *Complete linkage* considers the minimum inter-cluster pairwise similarity and thus merges clusters to produce a complete subgraph with respect to some threshold, connecting all edges between all nodes. Complete linkage does not take account of cluster structure and tends to produce compact clusters with equal diameters. It is most suitable when the true clusters are compact and roughly equal in size. *Mean linkage* considers the average inter-cluster pairwise similarity and tends to join clusters with small variances. It takes some account of cluster structure.

Our similarity data are subject to a substantial amount of noise caused by variations in image quality due to changing imaging conditions and the motion of the imaging system itself. Therefore, decisions based on individual similarity values such as in single and complete linkage are prone to producing unsatisfactory clusters. Even though this effect was ameliorated somewhat when using mean linkage, substantial classification errors remained.

3.1 Algorithm Outline

If we assume that workspaces often contain extended contiguous regions of similar character then images taken *in sequence* are more likely to depict a similar class of environment than images taken at random locations. Agglomerative clustering lends itself readily to the incorporation of such a *sequence constraint.* However, we found that, if an agglomerative clustering scheme is to be employed successfully on our data, a different similarity measure is required that takes account of cluster structure in terms of the number of pairwise connections gained by a possible merger.

Experiments further showed that the initial construction of highly consistent *seed* clusters drastically improves classification performance — this is justified since these seeds are representative of the final cluster characteristics and provide support when evaluating a similarity function based on cluster consistency. Adding a simple classification stage allows evaluation of the degree to which the resulting workspace classes are representative of the robot's environment. Thus, we propose the following sequence of processing steps (see Figure 2):

1. Construct consistent seed clusters from `M` using a sequence constraint.
2. Determine the similarity between existing (seed) clusters.
3. Starting with the seeds, construct workspace classes
   by iteratively merging suitably chosen clusters.
4. Classify all as yet unclassified images as belonging
   to one of the existing workspace classes.

Seed Cluster Construction A graph theoretical interpretation of a similarity matrix is attractive and it is tempting to equate the construction of highly consistent seed clusters to the finding of cliques. However, given the noise in the similarity

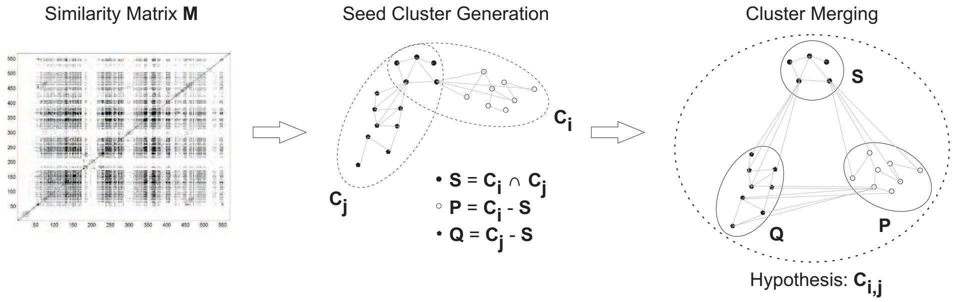

Fig. 2. From similarity matrix to workspace classes.

data, a strict construction of cliques tends to reduce the size of the seed clusters. This counteracts the supportive role that the seed clusters are intended to fulfil. Instead we propose the cluster seeding procedure given in Algorithm 1 in Table 1. In line 5, for every image i, a list of indices of similar images is drawn up and sorted according to similarity. The spatial constraint is introduced at this stage (line 8) by eliminating all members that are not part of a sequence. Thereby a certain leeway is given as to how many members may be missing from a sequence (three, in our case) to still be identified as being adjacent. The minimum length l_{min} of the resulting sequence is set to three for all experiments. Line 12 admits this seed cluster if image i is a member of this well connected, self similar set. Thus a set of highly consistent clusters is created which exploit the sequential nature of the data. Once all images have been considered the resulting seed clusters are inspected and clusters which form subsets of other clusters as well as carbon-copy clusters are eliminated. Typical resulting clusters are shown in Figure 3.

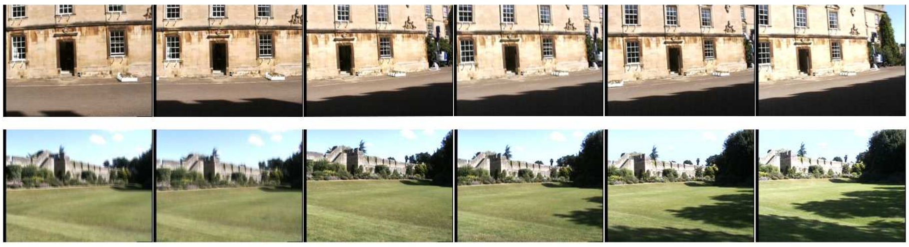

Fig. 3. Sample images from two seed clusters from the New College data set. These seeds were 'grown' into clusters 5 (top row) and 3 (bottom row) shown in Table 4.

Cluster Similarity We judge the similarity of two clusters by their inter-connectivity. It is tempting to adopt a strategy whereby only the connectivity of the resulting cluster is considered. However, with such an approach the connectivity of a relatively large cluster would dominate the merging decision. We consider only the connectivity of the intersection of the two sets and the inter-connectivity of their relative complements. With reference to the centre panel of Figure 2, two clusters $\mathbf{C}_i$ and $\mathbf{C}_j$ can be factored into their intersection, $\mathbf{S}$, and their complementary sets, $\mathbf{P}$ and $\mathbf{Q}$. Let $\mathbf{C}_{i,j}$ denote the union of $\mathbf{C}_i$ and $\mathbf{C}_j$ and $|\mathbf{C}_{i,j}|$ its cardinality N_m. The potential number of pairwise adjacent vertices (all potential pairs), N_E, in $\mathbf{C}_{i,j}$

Algorithm 1: SeedClusters

```
input: $\mathtt{M}_{N\times N}$ similarity matrix, min cluster size $c_{min}$, similarity threshold $s_{min}$, minimum sequence length $l_{min}$
output: seed clusters $\mathcal{C}_o = \{C_1, C_2, \cdots\}$
$\mathcal{C}_o \leftarrow \emptyset$
for $i = 1 : N$ do
    $\mathbf{S} \leftarrow \mathtt{SORT}(\mathtt{FIND}(\mathtt{M}[i,:] > s_{min}))$
    $C_i \leftarrow \emptyset$
    for $j = 1 : |\mathbf{S}|$ do
        if $\mathtt{FIND\_ADJACENT}(\mathbf{S}, \mathbf{S}[j]) \neq l_{min}$ then
            $C_i \leftarrow \{C_i, \mathbf{S}[j]\}$
        end if
    end for
    if $|C_i| > c_{min} \wedge i \cap C_i \neq \emptyset$ then
        $\mathcal{C}_o \leftarrow \{\mathcal{C}_o, C_i\}$
    end if
end for
```

Algorithm 2: MergeClusters

```
input: seed clusters $\mathcal{C}_o = \{C_1, \cdots C_K\}$
output: final clusters $\mathcal{C} = \{C_1, \cdots C_K\}$
repeat
    $B \leftarrow \mathtt{FALSE}$
    for $i = 1 : K$ ; $j = 1 : K$ do
        $\mathtt{D}(i,j) = d(\mathbf{C_i}, \mathbf{C_j})$
    end for
    for $i = 1 : K$ do
        $j^\star = \mathtt{MAXELEMENT}(\mathbf{D}(i,:))$
        if $\mathbf{D}(i, j^\star) > d_{min}$ then
            $i^\star = \mathtt{MAXELEMENT}(\mathbf{D}(:, j^\star))$
            if $i^\star == i$ then
                $\mathbf{C_i} \leftarrow \{\mathbf{C_i} \cup \mathbf{C_{j^\star}}\}$
                $\mathbf{C_{j^\star}} \leftarrow \emptyset$
                $B \leftarrow \mathtt{TRUE}$
            end if
        end if
    end for
until $!B$
```

Table 1. The outline of the two main algorithms of our approach is shown here, namely building seed clusters (Alg. 1) and merging (seed) clusters (Alg. 2).

can be expressed as

$$\binom{N_m}{2} = \overbrace{\binom{|\mathbf{S}|}{2} + \binom{|\mathbf{P}|}{2} + |\mathbf{S}| \cdot |\mathbf{P}|}^{\text{number of potential pairs in } \mathbf{C}_i} + \overbrace{\binom{|\mathbf{S}|}{2} + \binom{|\mathbf{Q}|}{2} + |\mathbf{S}| \cdot |\mathbf{Q}|}^{\text{number of potential pairs in } \mathbf{C}_j}$$

$$+ \overbrace{\binom{|\mathbf{P}|}{2} + \binom{|\mathbf{Q}|}{2} + |\mathbf{P}| \cdot |\mathbf{Q}|}^{\text{number of potential pairs in } \mathbf{P} \cup \mathbf{Q}} \overbrace{- \binom{|\mathbf{S}|}{2} - \binom{|\mathbf{P}|}{2} - \binom{|\mathbf{Q}|}{2}}^{\text{housekeeping terms}} \quad (1)$$

$$= \binom{|\mathbf{S}|}{2} + \binom{|\mathbf{P}|}{2} + \binom{|\mathbf{Q}|}{2} + |\mathbf{S}| \cdot |\mathbf{P}| + |\mathbf{S}| \cdot |\mathbf{Q}| + |\mathbf{P}| \cdot |\mathbf{Q}| \quad (2)$$

Based on Equation 2 and the considerations at the beginning of this subsection we obtain the similarity measure, $d(.)$, such that

$$d(\mathbf{C_i}, \mathbf{C_j}) = \frac{\binom{|\mathbf{S}|}{2} + \mid \mathbf{S} \mid \cdot \mid \mathbf{P} \mid + \mid \mathbf{S} \mid \cdot \mid \mathbf{Q} \mid + \epsilon}{\binom{N_m}{2} - \binom{|\mathbf{P}|}{2} - \binom{|\mathbf{Q}|}{2}} \quad (3)$$

where the term $|\mathbf{P}| \times |\mathbf{Q}|$ has been replaced with ϵ, denoting all pairwise adjacent vertices defined with respect to a threshold s_{min}.

The function $d(.)$ represents a monotonically increasing similarity function over the range $[0 \rightarrow 1]$. An additional advantage of the formulation in Equation 3 is an interpretation of $d(.)$ as the ratio of vertices added over potential vertices gained by a merging operation if sets **S**, **P** and **Q** were themselves complete subgraphs. Thus, in this case a threshold on the similarity function would specify a lower bound on the number of vertices in a merged cluster. However, in practice **S**, **P** and **Q** are well connected but not complete subgraphs. Despite this condition not being met empirical observations show $d(.)$ to provide a good approximation to such a lower bound.

Cluster Merging The merging procedure is given by Algorithm 2 in Table 1. The cluster similarity function d given in Equation 3 is used to construct a *cluster similarity matrix*, `D` (line 6). Considering each cluster $\mathbf{C}_i$ in turn, the corresponding maximally similar cluster, $\mathbf{C}_{j^\star}$ is identified in line 9. These two clusters are then merged if $\mathbf{C}_{j^\star}$ has $\mathbf{C}_i$ as its maximally similar cluster (lines 11-12). This merging criterion of mutual maximal similarity provides an effective termination for the merging phase.

Classification The classification scheme employed here is a naive nearest-neighbour classifier based on the mean-linkage criterion. A distance threshold is introduced below which images are assigned to a default 'unassigned' cluster.

4 Experimental Results And Discussion

In our experiments an autonomous vehicle equipped with onboard camera, odometry, 3D laser scanner and GPS (for ground truth) was driven around different workspaces. The sensor data was logged and processed offline although this was more for convenience than necessity. In fact, the processing of all the data sets shown (i.e. for about 1000 images) takes less than 30 seconds on a vanilla computer using Matlab. The proposed algorithm for scene classification has been evaluated for three different outdoor environments, "Thom Building", "New College", and "Cloister". The first is dominated by different kinds of buildings (modern and old), streets, and some lawns. The emphases of the second data set are grass fields, bushes, some buildings, and an ancient wall framing the area. The Cloister data set is a loop around 14th century walkways, containing close-ups of ancient brick walls and views into an enclosed quadrangle. In many ways this data set is similar to an indoor environment, though it has challenging lighting conditions and an uneven paved floor. Altogether the data sets cover about 1.5 kilometres of distance travelled.

Table 4 depicts sample images from clusters found in each data set. The classes found refer to either certain locations (cluster 3, Thom Building), more general concepts (both classes for Cloister) or prominent objects like parked cars (cluster 4, Thom Building). This is not to say that the algorithm in fact learns useful concepts of objects, but scenes where a certain object takes up most of the field of view are consistently clustered into the same class. A further example is the black container in cluster 1 (Thom Building). The most dominant class (in terms of size) in the New College data set is cluster 2 which can be broadly described by the terms "Bushes and Plants". This class also covers near-field views of crumbling walls, which in the absence of colour have similar texture. From this it seems that adding colour information to the feature descriptors would be beneficial. This point is the subject of our ongoing endeavour.

Some clusters represent rather similar concepts, like cluster 4 and 5, which capture a quadrangle of sandstone buildings. That the two clusters have not been merged is probably due to the features along the shadows in the scenes of cluster 5. However, obtaining several clusters for subjectively similar concepts does not lead to inconsistent labelings, as would be the case if one cluster presented two different concepts. For the resulting maps to be useful for human-machine communication, semantic labels would be manually assigned to the different clusters (similar to the column "Description" in Table 2). From this, the representation of concepts can be updated automatically by fusing clusters with very similar labels.

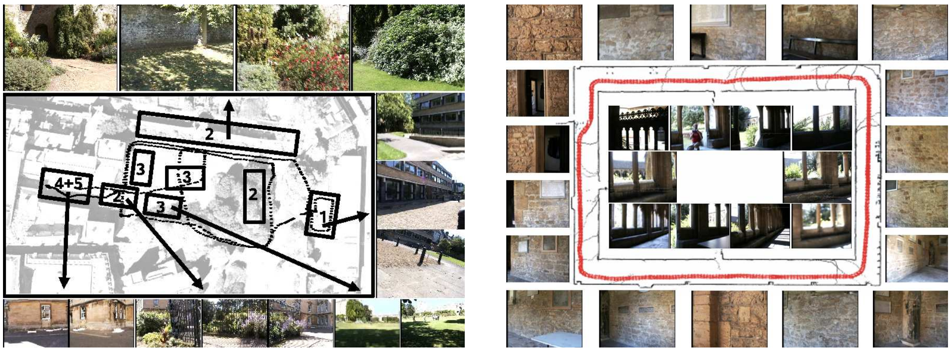

Fig. 4. This figure presents sample images from the clustered New College (left) and Cloister (right) data sets. Several different classes were identified for the diverse landscape of the New College gardens. Only two rather characteristic classes were learned for the more monotonous Cloister data set.

So far we have shown that the clustering results are plausible in the sense that clusters do refer to different prominent locations or common themes, although there might be several clusters describing a similar concept. The question is, to what extent can these clusters be considered consistent and how well are they supported by images that have not yet been assigned to any cluster. It is important to note that the main purpose of the algorithm presented here is to find consistent clusters, which may cover only a subset of the given data. Nevertheless, classifying the remaining scenes gives a good indication of the applicability of the learned concepts and their descriptive power.

Table 2 summarises some statistic properties for each of the generated clusters. *Minimum mean linkage (MML)* is the mean of the similarity values between one cluster member and all the others. Comparing the MML with the *mean linkage* value within one cluster implies that the respective scenes (last column in Table 4) seem to be rather dissimilar to most of the others in the cluster. This is plausible for cluster 5 (Thom Building) or cluster 2 (New College), though there are other similar images in these clusters (not depicted here). However, for most of the clusters this apparent discrepancy with respect to the MML value is not obvious. In fact, subjectively, these scenes seem to be very similar to the main theme of the cluster. This shows that despite considerable noise in the image similarities our algorithm manages to find consistent clusters. These clusters might not be fully connected (internally), i.e. they contain pairs with low similarity, but they do not comprise several "subclusters" that only share a single similar pair. This is the strength of the algorithm presented here. First, it exploits spatial constraints by generating seed clusters. Second, it introduces a cluster similarity measure in the merging step that considers the interconnectivity of clusters in a way that can be seen as a blend of complete and mean linkage. It approximates the number of additional outliers that would be caused by merging, and relates it to the number of possible connections and shared cluster members, see also Section 3.1. Thus, it favours strong interconnectivity like complete linkage, but at the same time tolerates a certain amount of outliers.

Thom Building Data Set - 387 images

Clst. ID	Description	size	mean lkg.	min mlkg.	inlier count
1	black container	27	0.39	0.03	254/351 = 0.72
6	close-up of metal fence	30	0.16	0.03	272/435 = 0.63
7	bicycle stand, traffic cone	49	0.19	0.06	548/1176 = 0.47
4	another black car	15	0.09	0.03	30/105 = 0.29
5	staircase, handrail, windows, black car	27	0.07	0.02	82/351 = 0.23
3	street crossing, houses, grass	21	0.08	0.04	42/210 = 0.20
2	grass, windows	34	0.06	0.03	78/561 = 0.14

New College Data Set - 570 images

Clst. ID	Description	size	mean lkg.	min mlkg.	inlier count
5	ancient building	22	0.20	0.09	171/231 = 0.74
4	same as 5, but in the shadow	13	0.11	0.05	24/78 = 0.31
3	ancient wall, grassland/field	28	0.10	0.04	96/378 = 0.25
1	modern house, windows, sand	19	0.09	0.07	32/171 = 0.19
2	mainly bushes & flowerbeds	128	0.09	0.04	1466/8128 = 0.18

Cloister Data Set - 212 images

Clst. ID	Description	size	mean lkg.	min mlkg.	inlier count
2	dark interior, bright windows	21	0.11	0.07	44/210 = 0.21
1	close up of ancient brick walls	14	0.10	0.08	18/91 = 0.20

Table 2. Clustering results for three data sets. The *Description* is based on subjective inspection of the resulting clusters (see also Table 4). *mean lkg.* is the mean of all similarity values between members of the cluster. *min mlkg.* is the minimum of the mean linkage of each member of the cluster with respect to the others. Finally, the inlier count is the ratio of all pairs within a cluster whose similarity value falls below s_{min} (see also Section 3.1) and the number of possible pairings within a cluster, i.e. $\binom{N}{2}$ where N is the size of the cluster.

Thom Building Data Set

Cluster ID	1	2	3	4	5	6	7
number of classified images	10	20	17	4	0	3	16
error (subjective)	0	0	0.06	0.25	-	0	0

New College Data Set

Cluster ID	1	2	3	4	5
number of classified images	7	86	3	4	6
error (subjective))	0	0.01	0	0	0.67

Cloister Data Set

Cluster ID	1	2
number of classified images	29	51
error (subjective)	0.1	0.26

Table 3. Results for classifying scenes that are not part of the cluster descriptions. Due to insufficient similarity some images are not classified, in particular 114 for the Thom Building data set, 254 for the New College data set and 97 for the Cloister data set.

5 Conclusions

In this paper we have presented a method for workspace labelling in the context of mobile robot mapping. The algorithm makes explicit use of the consecutive nature of the acquired data by forcing spatial constraints in a pre-clustering step. The resulting seed clusters are merged with respect to a cluster similarity measure that balances between the number of expected outliers, all possible connections and the number of common elements. Using an additional criterion, which we call mutual maximal similarity between clusters, the algorithm converges very fast. The algorithm has been evaluated on different outdoor data sets, and shown to produce plausible concepts of scenes. It is particularly robust to noise in the image similarities that arises from motion blur and variability of image quality.

References

1. I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," *Proceedings of International Conference on Robotics and Automation*, 2000.
2. P. Newman and K. Ho, "Outdoor SLAM using visual appearance and laser ranging," *IEEE International Conference on Robotics and Automation*, May 2006.
3. J. Porta and B. J. A. Kroese, "Appearance-based concurrent map building and localization," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 159–164, 2005.
4. G. Piatetsky-Shapiro, C. Matheus, P. Smyth, and R. Uthurusamy, "Kdd-93 - progress and challenges in knowledge discovery in databases," *Ai Magazine*, vol. 15, pp. 77–82, 1994.
5. M. W. Berry, *Survey of Text Mining: Clustering, Classification, and Retrieval.* Springer-Verlag, New York, 2003.
6. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
7. T. Hofmann, "Probabilistic Latent Semantic Analysis," in *Proc. of Uncertainty in Artificial Intelligence (UAI)*, 1999.
8. A. Bosch, A. Zisserman, and X. Munoz, "Scene Classification via pLSA," in *Proc. of the European Conference on Computer Vision*, 2006.
9. R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning Object Categories from Google's Image Search," in *Proc. of the Int. Conference on Computer Vision*, 2005.
10. J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of the Int. Conference on Computer Vision*, Nice, France, Oct. 2003.
11. T. M. Mitchell, *Machine Learning.* McGraw-Hill Science/Engineering/Math, 1997.
12. L. Fei-Fei, R. Fergus, and P. Perona, "A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories," in *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, 2003, pp. 1134–1141.
13. C. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, no. 1, pp. 63–86, 2004.
14. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th International Conference on Computer Vision, Kerkyra*, 1999, pp. 1150–1157.
15. K. S. Jones, "Exhaustivity and specificity," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.
16. C. Stachniss, O. Martínez-Mozos, A. Rottmann, and W. Burgard, "Semantic labeling of places," in *Proceedings of the International Symposium on Robotics Research*, San Francisco, CA, USA, 2005.
17. J. B. Kruskal and M. Wish, "Multidimensional scaling," *Sage University Paper series on Quantitative Applications in the Social Sciences*, no. 07-011, 1978.
18. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
19. B. D. Ripley, *Pattern recognition and neural networks.* Cambridge University Press, 1996.
20. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition).* Wiley-Interscience, 2000.
21. B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis.* Oxford University Press Inc., New York, NY, 2001.

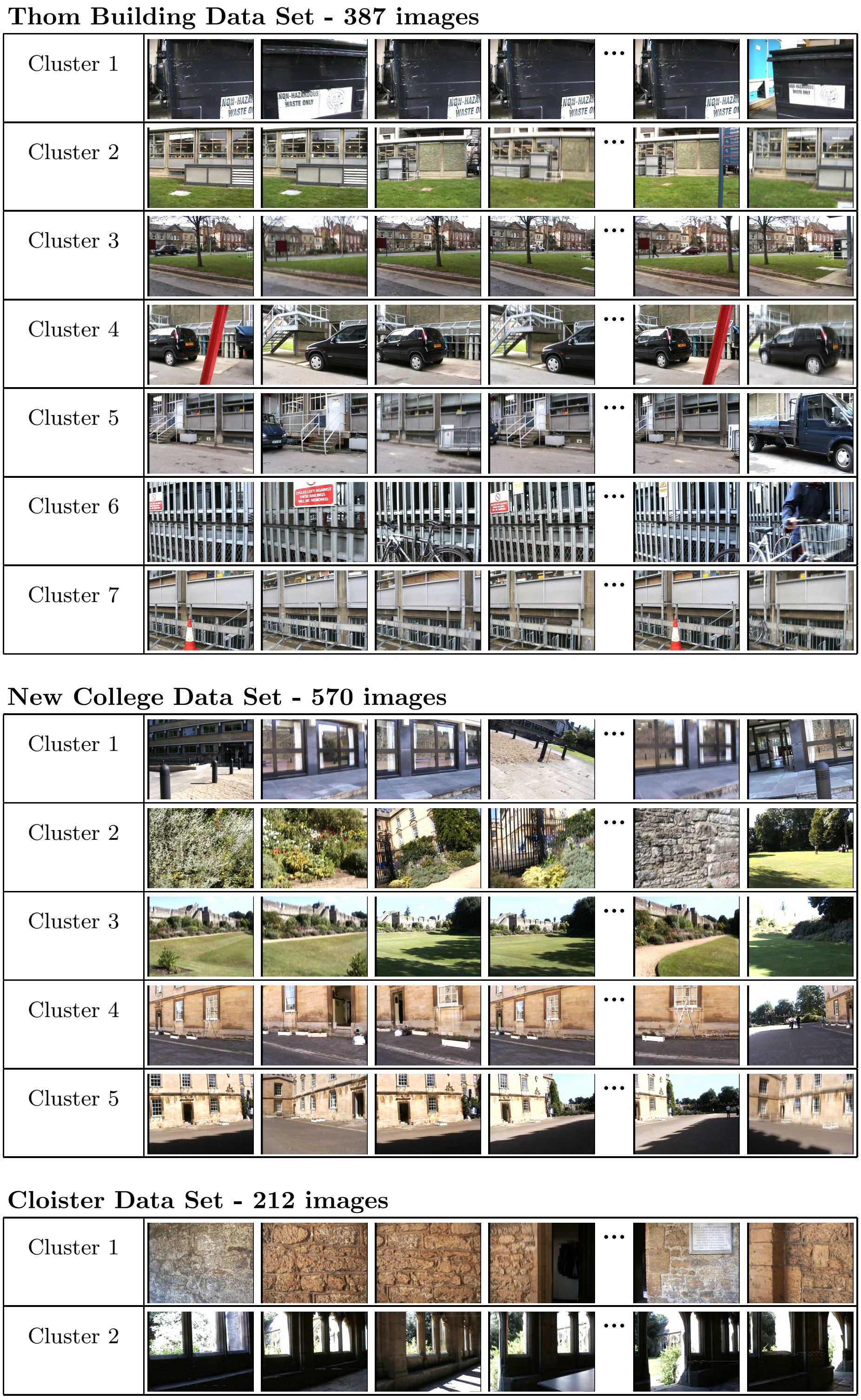

Table 4. Images for all clusters (and data sets) as presented in Table 2. Note that some clusters seem to describe particular locations within the environment, e.g. cluster 3 of the Thom Building data set. Others refer to more general concepts (Cloister data set, cluster 2 and 3 of the New College data set). And some seem to be dominated by prominent objects like parked cars or containers (cluster 1 and 4 of the Thom Building data set).