Probabilistic Appearance Based Navigation and Loop Closing

Mark Cummins and Paul Newman

Oxford University Mobile Robotics Research Group. {mjc,pnewman}@robots.ox.ac.uk

Abstract— This paper describes a probabilistic framework for navigation using only appearance data. By learning a generative model of appearance, we can compute not only the similarity of two observations, but also the probability that they originate from the same location, and hence compute a pdf over observer location. We do not limit ourselves to the kidnapped robot problem (localizing in a known map), but admit the possibility that observations may come from previously unvisited places. The principled probabilistic approach we develop allows us to explicitly account for the perceptual aliasing in the environment – identical but indistinctive observations receive a low probability of having come from the same place. Our algorithm complexity is linear in the number of places, and is particularly suitable for online loop closure detection in mobile robotics.

I. INTRODUCTION

This paper presents a Bayesian framework for localization and mapping using appearance-based observations of the local scene. The measurements we use are collections of "words", each representing some small aspect of the local workspace as captured by the sensor — typically, but not necessarily, a camera. The underlying map representation is a set of discrete places each parameterized by a probability distribution over the existence of words at that place. We show how by learning a generative observation model for scene appearance we are able to probabilistically localize the observer and update place models with a time complexity which is linear in the number of places mapped. We apply our method to the exploration (1.6 km driven path length) of an initially unknown environment, capturing the local workspace using camera images. In particular we highlight the computational efficiency of our approach, the extremely low error rate in place matching and the linear time ability to detect loop closure.

Recently, promising results have been obtained with appearance based approaches to loop closure [3], [4], [5], [6], [7], which draw on image-matching techniques developed in the computer vision community [8]. Typically, existing methods are based on a similarity measure between current and past observations. Using similarity for loop closure is complicated by perceptual aliasing. Widely separated areas of the environment may appear very similar, even to a discriminative sensor such as a camera. In [3], this problem was addressed by using an SVD-based decomposition of the similarity matrix to suppress matches due to repetitive environmental features. While this method yields good results,

its time complexity is cubic in the number of observations, making it infeasible to apply online.

A key idea in this paper is that for navigation, we are interested not in the similarity of two observations, but rather in the probability that the observations come from the same place (for example see Figure 3). Two identical but indistinctive observations may indicate a low probability of being in the same place. We construct a Bayesian framework for assessing this probability directly, which obviates the need for any secondary method to deal with the effect of perceptual aliasing, and results in an efficient scheme suitable for online application. Our method also has the advantage that it makes full use of the available information in the measurements, including the negative observations.

II. REPRESENTING APPEARANCE

We will model the world as a set of discrete and disjoint places. For each new observation that the vehicle collects, we calculate the probability that the observation came from one of the known places or a new place, and then update our belief about the appearance of each place.

We adopt a "bag of words" representation of raw sensor data [8], where scenes are represented as collection of attributes (words) chosen from a set (vocabulary) of size $|v|$. An observation of local scene appearance, visual or otherwise, captured at time k is denoted $Z_k = \{z_1, \ldots, z_{|v|}\}$, where z_i is a binary variable indicating the presence (or absence) of the ith word of the vocabulary. Furthermore $\mathcal{Z}^k$ is used to represent the set of all observations up to time k.

The results in this paper employ binary features derived from imagery, based on quantized SIFT descriptors (see Section IV); however binary features from any sensor or combination of sensors could be used. For example, quantized versions the features proposed in [11], [12], [13] might be used for laser or sonar data.

We shall begin by learning a generative model of the observations from a training set of typical sensor data. This is a one-off process that occurs offline, and is described in Section II-A. The process of matching observations to places is then presented in section III.

A. Learning an Observation Model

In order to use our bag-of-words representation in a probabilistic framework, a generative model of the observations is needed. Not all words are equally distinctive, and word occurrence is far from independent. Computing the full distribution is infeasible for any vocabulary of reasonable size; instead we compute a thin junction tree approximation
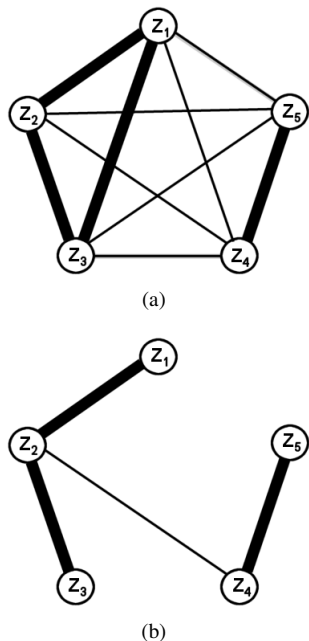
(a)

(b)

Fig. 1. **(a)** The complete mutual information graph. Darker edges indicate more mutual information. **(b)** A maximum weight spanning tree. The best approximation (smallest KL divergence) to $p(Z)$ involving only first-order conditionals is given by $p(z_2)p(z_1|z_2)p(z_3|z_2)p(z_4|z_2)p(z_5|z_4)$.

to the full distribution [2]. In particular, we have chosen to use a width-one junction tree as described by Chow and Liu [1]. The Chow Liu tree has the advantage that it can be efficiently computed even for large vocabulary sizes, is guaranteed to be the optimal approximation within its model class, and requires only first order conditional probabilities, which can be reliably estimated from available training data.

If $Q(Z)$ is a distribution on n discrete variables, $Z = \{z_1, z_2, \ldots, z_n\}$, then the Chow Liu algorithm constructs a tree-structured Bayesian network which represents a distribution $P(Z)$ approximating the true distribution $Q(Z)$ in the sense of minimizing the Kullback-Leibler divergence

$$D_{KL}(Q, P) = \sum_Z Q(Z) log \frac{Q(Z)}{P(Z)} \qquad (1)$$

The Chow Liu tree is a maximum weight spanning tree of the mutual information graph. It is constructed by first computing a complete graph of n nodes (one for each z_i) , where each edge (z_i, z_j) has weight equal to the mutual information $I(z_i, z_j)$ between variable i and j. The mutual information is given by

$$I(z_i, z_j) = \sum_{z_i \in \Omega, z_j \in \Omega} p(z_i, z_j) log \frac{p(z_i, z_j)}{p(z_i)p(z_j)} \qquad (2)$$

and the summation is carried out over all possible states of z – in our binary case this implies $\Omega = \{0, 1\}$. Mutual information measures the degree to which knowledge of the value of one variable predicts the value of another. It is zero if two variables are independent, and strictly larger otherwise.

The Chow Liu algorithm reduces the complete dependence graph to a tree, retaining a maximum weight set of edges that

meet the tree constraint (see Figure 1). The dependencies that we approximate to independent have as little mutual information as possible. This guarantees the optimal tree-structured approximation to the true distribution.

A tree-structured approximation is particularly useful because we can express the joint as a product of first-order conditionals (the edges in the tree). Thus

$$p(Z) = p(z_1, \ldots, z_n) = p(z_r) \prod_{i=1}^n p(z_i|z_{p_i}) \qquad (3)$$

where z_r is the root of the tree and z_{p_i} is the parent of z_i in the tree. For example, in Figure 1, the full distribution

$$
\begin{aligned}
p(Z) \quad = \quad & p(z_1|z_2, z_3, z_4, z_5)p(z_2|z_3, z_4, z_5) \\
& p(z_3|z_4, z_5)p(z_4|z_5)p(z_5) \qquad (4)
\end{aligned}
$$

is approximated as

$$p(Z) \approx p(z_2)p(z_1|z_2)p(z_3|z_2)p(z_4|z_2)p(z_5|z_4) \qquad (5)$$

A concrete example of how this applies to our image data is described in Section IV-A.

Typically the computation of the Chow Liu tree is achieved directly by computing the mutual information graph from data and then finding the maximum weight spanning tree[1], but for sparse data such as in our case, it can be accelerated using an algorithm described by Meilă [15].

The maximum likelihood estimates of the conditional probabilities $p(z_i|z_j)$ required in the expansion can be obtained directly from co-occurrence frequency in the training data – we can now calculate $p(Z)$. To mitigate potential problems due to limited training data, it may be necessary to use smoothing techniques such as the pseudo-Bayesian p^* estimator described in [16], rather than the maximum likelihood estimator. This prevents any probabilities from having unrealistic values 0 or 1.

III. REPRESENTING LOCATION

Given a vocabulary and an approximate probabilistic model of observations, it is now possible to compute a pdf over location from an observation. At time k, we model the world as collection of n_k discrete and disjoint locations $\mathcal{L}^k = \{L_1, \ldots, L_{n_k}\}$. We will denote the event that a word exists e_i, and the event that the existence of a word is reported by our detector as z_i. Each location L_i is described by the set $\{p(e_1 = 1|L_i), \ldots, p(e_{|v|} = 1|L_i)\}$, an estimate of the probability that each word exists at the location. A goal of this paper is to compute $p(L_i|\mathcal{Z}^k)$. The final ingredient necessary before we can do this is a model of our detector – the probability that a word is observed, given that it doesn't exist $p(z_i = 1|e_i = 0)$ and the probability that a word is not observed given that it actually exists $p(z_i = 0|e_i = 1)$. [2]

---

[1]For large vocabularies, it may require a semi-external spanning tree algorithm [14].

[2]Detector error rates can be set on a per-feature basis. For example, features associated with cars might exhibit a higher apparent error rate than features associated with buildings. However, in practice we have found that when the features are generated by a single type of sensor, good results can be obtained by simply setting the error rate to a single value for all words.

These two detector terms, plus a prior term that sets the probability that a new observation comes from a previously unknown location, are the only user-specified inputs to the algorithm. Of these, the algorithm is particularly sensitive only to the detector terms, which can be determined from a simple calibration described in Section IV-B.

A. Estimating Location via Recursive Bayes

Calculating $p(L_i|\mathcal{Z}^k)$ can be formulated as the recursive Bayes estimation problem:

$$p(L_i|\mathcal{Z}^k) = \frac{p(Z_k|L_i)p(L_i|\mathcal{Z}^{k-1})}{p(Z_k|\mathcal{Z}^{k-1})} \quad (6)$$

Here $p(L_i|\mathcal{Z}^{k-1})$ is our prior belief about our location (which we obtain from our previous position estimate modified by a motion model, using the assumption that sequential observations were taken at adjacent places), $p(Z_k|\mathcal{Z}^{k-1})$ is a normalizing term, and the observation likelihood $p(Z_k|L_i)$ can be expanded using the Chow-Liu approximation to obtain:

$$p(L_i|\mathcal{Z}^k) \approx \frac{p(L_i|\mathcal{Z}^{k-1})p(z_r|L_i)\prod_{q=1}^{|v|}p(z_q|z_{p_q},L_i)}{p(Z_k|\mathcal{Z}^{k-1})} \quad (7)$$

The observation factors in Equation 7 can be further expanded as:

$$p(z_q|z_{p_q},L_i) =$$

$$\sum_{s_{e_q}\in\{0,1\}} p(z_q|e_q = s_{e_q}, z_{p_q}, L_i)p(e_q = s_{e_q}|z_{p_q}, L_i) \quad (8)$$

which, making the approximation that detector errors are independent of position in the world, and that $p(e_j)$ is independent of z_i for all $i \neq j$, becomes:

$$p(z_q|z_{p_q},L_i) = \sum_{s_{e_q}\in\{0,1\}} p(z_q|e_q = s_{e_q}, z_{p_q})p(e_q = s_{e_q}|L_i)$$
$$(9)$$

The term $p(z_q|e_q, z_{p_q})$ can be expanded as

$$p(z_q = s_{z_q}|e_q = s_{e_q}, z_p = s_{z_p}) = (1 + \frac{\alpha}{\beta})^{-1} \quad (10)$$

where $s_{z_q}, s_{e_q}, s_{z_p} \in \{0,1\}$ and

$$\alpha = p(z_q = s_{z_q})p(z_q = \overline{s_{z_q}}|e_q = s_{e_q})p(z_q = \overline{s_{z_q}}|z_p = s_{z_p})$$

$$\beta = p(z_q = \overline{s_{z_q}})p(z_q = s_{z_q}|e_q = s_{e_q})p(z_q = s_{z_q}|z_p = s_{z_p})$$

α and β are now expressed entirely in terms of quantities which can be estimated from training data. Hence we can now calculate $p(Z_k|L_i)$.

B. New Place or Old Place?

We now turn our attention to the denominator of Equation 7, $p(Z_k|\mathcal{Z}^{k-1})$, which is key to determining whether the observation comes from a new or previously visited location. For pure localization we can compute a pdf over location simply by normalizing the observation likelihoods computed as described in the previous section. However, we would like to admit the possibility that a new observation comes from a previously unknown location, which requires an explicit calculation of $p(Z_k|\mathcal{Z}^{k-1})$.

If we divide the world into the set of mapped places M and the unmapped places $\overline{M}$, then

$$p(Z_k|\mathcal{Z}^{k-1}) = \sum_{m\in M} p(Z_k|L_m)p(L_m|\mathcal{Z}^{k-1})$$
$$+ \sum_{n\in\overline{M}} p(Z_k|L_n)p(L_n|\mathcal{Z}^{k-1}) \quad (11)$$

where we have assumed observations are conditionally independent given location. We cannot evaluate the second summation directly, because it involves all possible unknown places, and additionally the prior terms are unknown. However, the term can be approximated by a sampling procedure. To do this, we create random place models L_n by sampling observations from the Chow-Liu tree (sampling from the tree is simple and extremely rapid). The likelihood $p(Z_k|L_n)$ for each of these places can then be evaluated. The prior terms we set uniformly among the samples, with the overall prior that a new image comes from an unmapped place being user-defined.

The intuitive explanation of what is being calculated here is as follows: if the current observation Z_k is distinctive, and a good match for one of the mapped places, it is very unlikely that we will randomly sample a place model with a higher likelihood of having generated the observation. On the other hand, if Z_k is a common observation or a poor match for all mapped places, then the chances of sampling a place model with equal or higher likelihood of having generated it are high. This means a higher value of the denominator $p(Z_k|\mathcal{Z}^{k-1})$ (and thus a lower probability that the current observation came from a mapped place).

Before we can evaluate $p(L_i|\mathcal{Z}^k)$, we finally have to describe the creation of our place appearance models $\{p(e_1 = 1|L_j), \ldots, p(e_{|v|} = 1|L_j)\}$. These are the only terms mentioned in Equation 7 which we are not yet able to evaluate. This is addressed in the next section.

C. Updating Place Models

The term $p(e_i = 1|L_j)$ represents the probability of the existence of word i at place j. A place is initialized so that all words exist with marginal probability $p(e_i = 1)$. When we have an observation that relates to the place, we can update our beliefs via

$$p(e_i = 1|L_j, \mathcal{Z}^k) =$$

$$\frac{p(Z_k|e_i, L_j)p(e_i|L_j, \mathcal{Z}^{k-1})}{\sum_{s_e\in\{0,1\}} p(Z_k|e_i = s_e, L_j, \mathcal{Z}^{k-1})p(e_i = s_e|L_j, \mathcal{Z}^{k-1})}$$
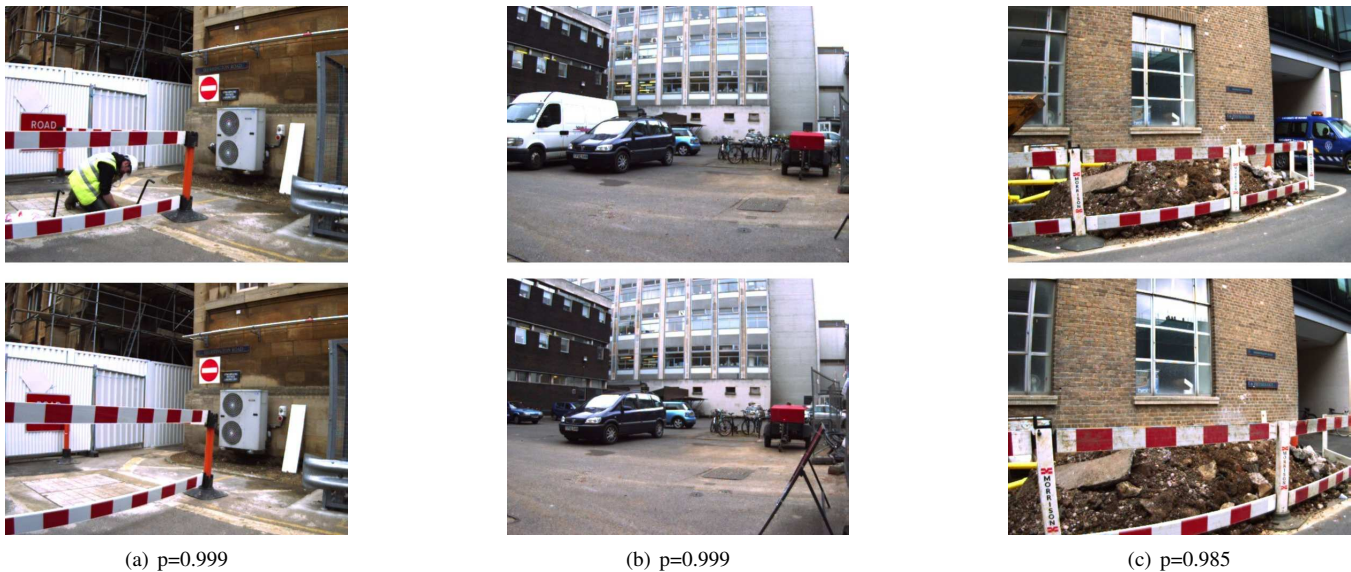$$(12)$$

(a) p=0.999   (b) p=0.999   (c) p=0.985

Fig. 2. Some examples of images that were assigned high probability of having come from the same place. Notice that the algorithm is tolerant of scene change when the images contain distinctive features. The probability that the two images come from the same place is indicated below the pairs.

(a) p=0.416   (b) p=0.699   (c) p=0.30

Fig. 3. Some remarkably similar-looking images from different parts of the workspace. The algorithm correctly assigns them low probability of having come from the same place, because our generative model has correctly learned that the words in these images are highly correlated, and hence the images have low distinctiveness. The probability that the two images come from the same place is indicated below the pairs

We make the approximations that $p(e_i|Z) \approx p(e_i|z_i)$, and $p(z_i|e_i, L_j) \approx p(z_i|e_i)$ leading to

$$p(e_i = 1|L_j, \mathcal{Z}^k) =$$

$$\frac{p(z_i|e_i)p(e_i|L_j, \mathcal{Z}^{k-1})}{\sum_{s_e \in \{0,1\}} p(z_i|e_i = s_e)p(e_i = s_e|L_j, \mathcal{Z}^{k-1})} \quad (13)$$

Note that this simplification equates to only updating the existence of word i at a place using observations relating to word i. We do not attempt to make inferences about the existence of word i from observations of other words.

Finally we must address the issue of relating a particular observation to a place (new or otherwise). This is a data association problem. Currently, we use an ML decision - if the maximum likelihood location is a new place, we add a new location to our map, initializing it so that all words exist with marginal probability $p(e_i|L_{new}) = p(e_i)$, and then apply the update in Equation 13. It would also be possible to adopt a more sophisticated approach, such as maintaining a pdf over topologies using a particle filter, as described in [4].

D. Summary of Approximations and Parameters

We briefly recap the approximations and user-defined terms in our probabilistic scheme.

Approximations:

1) Observations are conditionally independent given position:

$$p(Z_k|L_i, \mathcal{Z}^{k-1}) \approx p(Z_k|L_i)$$

This approximation will be very close to the truth for most sensors.

2) Detector errors are independent of position:

$$p(z_j|e_j, L_i) \approx p(z_j|e_j)$$

This approximation also seems unlikely to introduce significant error.

3) Observations of one feature do not inform us about the existence of other features:

$$p(e_j|Z_k) \approx p(e_j|z_{j_k})$$

In effect this simplification means that we will be somewhat less certain about feature existence than if we made full use of the available data. However, it would seem unlikely to cause problems.

4) Computation of normalization factor $p(Z_k|\mathcal{Z}^{k-1})$ approximated by sampling.

Input Parameters:

1) Detector model (two scalars):
   $p(z_i = 1|e_i = 0)$ and $p(z_i = 0|e_i = 1)$

2) Prior that, while exploring, an new observation comes from a new place, which we used to compute:
   $\sum_{n \in \overline{M}} p(L_n|\mathcal{Z}^{k-1})$

IV. RESULTS

We tested the described algorithm using imagery from a mobile robot. Each image that the robot collects is passed into a processing pipeline that produces a bag-of-words representation, which is then the input to the algorithm described in the previous section. Producing the bag-of-words representation could be achieved in many ways. In our implementation we extract regions of interest from the images using the Harris-affine detector [10] then compute SIFT descriptors [9] around the regions of interest. Finally, we map regions of SIFT space to visual words as suggested in [8]. This is achieved by clustering all the descriptors from a set of training images using a simple agglomerative procedure, then quantizing each SIFT in the test images to its approximate nearest cluster centre using a kd-tree.

A. Building the Vocabulary Model

The next step is to construct a Chow-Liu tree to capture the co-occurrence statistics of the visual words. To do this, we construct the mutual information graph as described in Section II-A. Each node in the graph corresponds to a visual word, and the edge weights (mutual information) between node *i* and *j* are calculated as per Equation 2 - essentially this amounts to counting the number of images in which word *i* and *j* co-occur. The Chow-Liu tree is then the maximum spanning tree of the graph.

If the Chow-Liu tree is to be a good approximation to the true distribution, it must be computed from a large number of observations. To prevent bias, these observations should be independent samples from the observation distribution. We collected three thousand images from 30 km of urban streets using a camera mounted on the side of a truck (it was too far to walk). The images were taken 10m apart, perpendicular to the truck's motion, so that they are non-overlapping and approximate independent samples from the observation distribution. From this dataset we compute the clustering of SIFT features that form our vocabulary, then compute the Chow Liu tree for the vocabulary. The clustering procedure generated a vocabulary of approximately 35k words. The combined process takes approximately 3 hours on a 3GHz Pentium IV.

B. Performance

We used this vocabulary to navigate using imagery collected by a mobile robot. The robot travelled over an outdoor trajectory of 1.6km, collecting 1735 images. The number of samples for determining the denominator term $p(Z_k|\mathcal{Z}^{k-1})$ was set equal to twice the number of places currently in the map, subject to a minimum of 100 samples. We modelled our detector by $p(z_i = 0|e_i = 1) = 0.4$ and $p(z_i = 1|e_i = 0) = 0$ – i.e. a per-word false negative rate of 40%, and no false positives[3]. Note that the sensor has a high false negative rate (due to the performance of the Harris-affine region detector). In addition, the imagery collected by the robot differs considerably from the training images - the robot imagery was collected in a different area of the city, and the robot's camera is much closer to the ground than the truck's, meaning that the typical features in the test imagery differ considerably with respect to the training set. However, our probabilistic scheme is still able to navigate effectively.

Figure 2 shows some typical images to which the algorithm assigns high probability of originating from the same location. Figure 3 highlights robustness to perceptual aliasing — the algorithm correctly assigns the images low probability of having come from the same location, despite strong visual similarity. This is because our generative observation model has correctly learned that the features in these images are highly correlated, thus the images are not distinctive.

The overall performance of the algorithm is presented in Figures 4, 5 and 6. Notice in Figure 4 that the algorithm correctly determines its location with high confidence ($p \geq 0.999$) in substantial portions of the loop closure. There are no false matches with comparable confidence. The algorithm run time per update is linear in the number of mapped places. The maximum time required to update the pdf (at the end of the run) was 6 seconds on a 3GHz Pentium IV. As the robot takes images approximately every 1.5 seconds, this is close to real time performance.

---

[3]To set these parameters, we first assumed a false-positive rate of zero. The false negative rate could then be determined by collecting multiple images at a test location $- p(z_i = 0|e_i = 1)$ then comes directly from the ratio of the number of words detected in any one image to the number of unique words detected in union of all the images from that location.
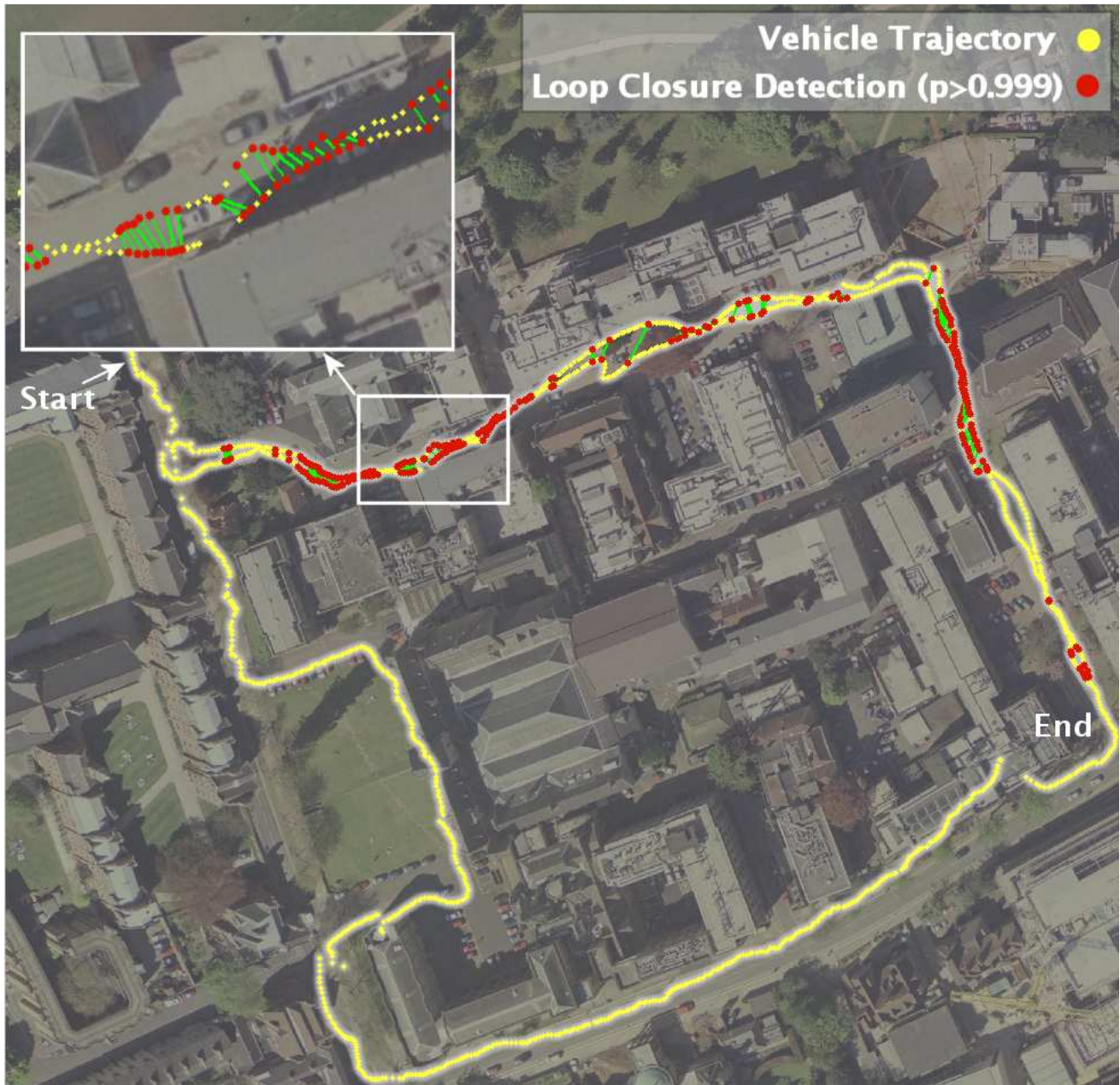
Fig. 4. Appearance based matching results overlaid on an aerial photograph. The robot travels clockwise around a loop with total path length 1.6km, beginning in the top left and finishing in the bottom right. The loop closure comprises two sides of the square. Positions (from GPS) at which the robot collected an image are marked with a yellow dot. Two images that were assigned a probability $p \geq 0.999$ of having come from the same location (on the basis of appearance alone) are marked in red and joined with a line. There are no incorrect matches at this confidence level.
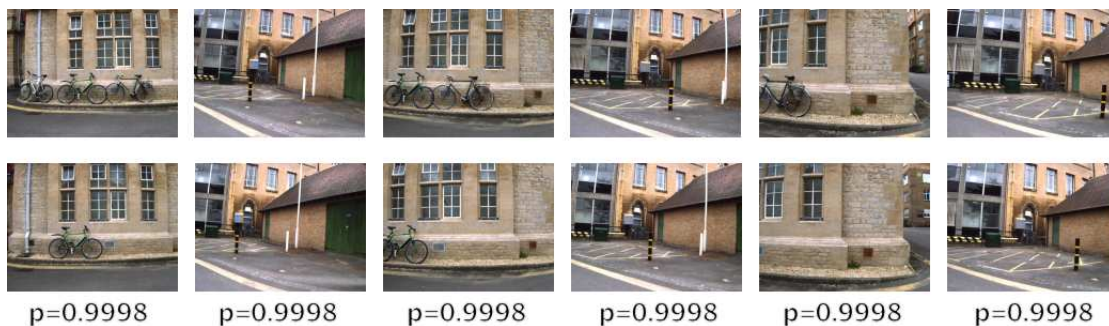

p=0.9998   p=0.9998   p=0.9998   p=0.9998   p=0.9998   p=0.9998

Fig. 5. A sample of the images that form the loop closure, with associated probabilities.

<div align="center">p=0.9998   p=0.9973   p=0.9998   p=0.6944   p=0.9995   p=0.9998</div>
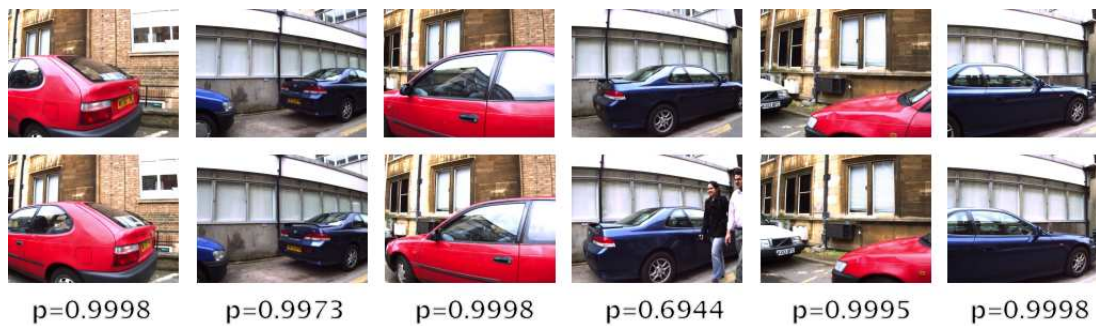
Fig. 6. Another sample of images, only a portion of which meet the $p \geq 0.999$ loop-closure threshold. In the fourth image in this sequence the loop closure probability drops significantly – this is because over half the interest points detected in the image are due to the pedestrians, thus the words in the two images differ considerably.

V. Conclusions and Future Work

We have presented a probabilistic framework for location inference using only appearance measurements. The approach is robust to perceptual aliasing and makes full use of the available observations, including negative observations. Additionally it has time complexity linear in the number of mapped locations, and is trivial to parallelize. We have demonstrated its use for loop closure detection on a mobile robot using only appearance data.

Several questions remain open for future research. With respect to our core algorithm, these is scope for improved handling of data association. It would also be interesting to investigate the use of junction trees of wider tree-width for representing our observation distribution.

More broadly, developing this algorithm has strongly highlighted the shortcomings of passive perception. While we successfully assign low probability to a large fraction of potentially confusing images such as those illustrated in Figure 3, we might question why the robot is collecting such poor imagery in the first place, as it is not useful for navigation. It would be interesting to improve the quality of the imagery collected by applying our probabilistic observation model to guide active perception.

It would also be interesting to learn detector models on a per-word basis. This would allow us to assign more weight to words associated with stable features such as buildings, and relatively less weight to words associated with transient objects. This should improve robustness to scene change (e.g Fig. 6).

Acknowledgments

References

[1] C.K. Chow and C.N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees", *IEEE Transactions on Information Theory*, vol. IT-14, No. 3, May 1968.

[2] F.R. Bach and M.I. Jordan, "Thin Junction Trees", *Advances in Neural Information Processing Systems*, 2002

[3] P. Newman, D. Cole, K. Ho, "Outdoor SLAM using Visual Appearance and Laser Ranging", *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, Florida, USA, May 15-19 2006.

[4] A. Ranganathan, F. Dellaert, "A Rao-Blackwellized Particle Filter for Topological Mapping", *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, Florida, USA, May 15-19 2006.

[5] A. Levin and R. Szeliski, "Visual Odometry and Map Correlation", *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004

[6] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments", *IEEE Intl. Symposium on Computational Intelligence in Robotics and Automation (CIRA),* 318-325, November 1999.

[7] C. Silpa-Anan and R. Hartley, "Visual localization and loop-back detection with a high resolution omnidirectional camera", *Workshop on Omnidirectional Vision*, 2005

[8] J Sivic, A Zisserman, "Video Google: a text retrieval approach to object matching in videos", *Proc. of the Intl. Conf. on Computer Vision*, Nice, France, Oct. 2003

[9] D.G. Lowe, "Object recognition from local scale-invariant features", *Proc. of the 7th Intl. Conf. on Computer Vision*, Kerkyra, 1999, pp. 1150-1157

[10] K. Mikolajczyk, C. Schmid, "Scale & Affine Invariant Interest Point Detectors", *International Journal of Computer Vision*, no. 1, pp. 63-86, 2004

[11] A. Frome, D. Huber, R. Kolluri, T. Bulow and J. Malik, "Recognizing objects in range data using regional point descriptors", *Proc. European Conf. on Computer Vision*, Springer, 2004

[12] O. Mozos, C. Stachniss and W. Burgard, "Supervised Learning of Places from Range Data using AdaBoost", *Proc. of the 2005 IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2005, 1730-1735

[13] A. Johnson, "Spin-Images: A Representation for 3-D Surface Matching", PhD Thesis, Carnegie Mellon University, 1997.

[14] R. Dementiev, P. Sanders, D. Schultes, J. Sibeyn, "Engineering an External Memory Minimum Spanning Tree Algorithm", *Proc. 3rd Intl. Conf. on Theoretical Computer Science*, 2004

[15] M. Meilă, "An Accelerated Chow and Liu Algorithm: Fitting Tree Distributions to High Dimensional Sparse Data", *Proc. 16th Intl. Conf. on Machine Learning*, 1999.

[16] Y.M.M. Bishop, S.E. Fienberg, P.W. Holland, *Discrete Multivariate Analysis - Theory and Practice*, MIT Press, Cambridge, Massachusetts, 1975