Accelerating FAB-MAP With Concentration Inequalities

Mark Cummins and Paul Newman

***Abstract*—We outline an approach for using concentration inequalities to perform rapid approximate multi-hypothesis testing. In a scenario where multiple hypotheses are ranked according to a large set of features, our scheme improves the efficiency of selecting the best hypothesis by providing a "bail-out threshold" at which unpromising hypotheses can be excluded from further evaluation. We show how concentration inequalities can be used to derive principled bail-out thresholds, subject to a user-specified error tolerance. The technique is similar to the sequential probability ratio test, but is applicable in more general conditions. We apply the technique to improve the speed of the fast-appearance-based mapping system for appearance-based place recognition and mapping. The speed increase provided by the new approach is data dependent, but we demonstrate speed improvements of between 25x − 50x on real data, with only a slight degradation in accuracy.**

***Index Terms*—Computer vision, recognition, simultaneous localization and mapping (SLAM).**

I. Introduction

The MOTIVATION of this paper is to improve the speed of our fast-appearance-based mapping system (FAB-MAP), which is an appearance-based navigation system for mobile robots [5], [7]. The FAB-MAP system allows a robot to incrementally construct an "appearance map" of its environment, which consists of a set of discrete locations, each with an associated appearance model. Distinctive places can be recognized even after unknown vehicle motion, and therefore, the appearance map allows the robot to perform loop-closure detection and other challenging global relocalization tasks in cases where metric methods for simultaneous localization and mapping (SLAM) may have failed.

The basic FAB-MAP model has fairly high computational cost. When real-time place recognition is required, its applicability is limited to maps that contain around 1000 locations (or, equivalently, a robot trajectory no more than about 1 km long). The limiting computational cost of the method is the calculation of an observation likelihood term in the probabilistic model. Every time the robot collects a new observation, this appearance likelihood term must be evaluated for all locations in the map. However, only a small number of these locations will yield non-negligible probability of having generated the observation. The core idea of this paper is that it should be possible to identify locations that will have insignificant likelihood before the calculation is fully complete. These locations could then be excluded from further processing, and large speed increases could be realized. We will describe a new multi-hypothesis testing technique, which formalizes this intuition and provides a closed-form bound on the resultant error rate.

Manuscript received May 23, 2010; revised September 14, 2010; accepted September 18, 2010. Date of publication November 1, 2010; date of current version December 8, 2010. This paper was recommended for publication by Associate Editor C. Stachniss and Editor W. K. Chung upon evaluation of the reviewers' comments. This work was supported by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the U.K. Ministry of Defence and by the EPSRC.

The authors are with the Mobile Robotics Research Group, Oxford University, OX1 3PJ Oxford, U.K. (e-mail: mjc@robots.ox.ac.uk; pnewman@robots.ox.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2010.2080390

II. Related Work

The problem of efficient multi-hypothesis testing is very generic and arises in many fields. One powerful approach for pruning a large space of hypotheses is the branch-and-bound technique, as used, for example, in the recent work of Lampert *et al.* to accelerate sliding window classification [10]. A classical branch-and-bound algorithm removes a hypothesis from consideration only when it is completely certain that the hypothesis cannot be the best solution. The method we present here can be thought of as a relaxation of this approach to the probabilistic case, where a hypothesis is removed when it is extremely likely (but not certain) that it is not the best hypothesis. By allowing for a small probability of error, we can achieve a much greater speed increase.

Such probabilistic "bail-out" strategies have been described elsewhere in computer vision, notably in the context of efficient RANSAC algorithms [12], [13]. Matas and Chum showed that for RANSAC, the sequential probability ratio test (SPRT) yields the optimal solution. The SPRT approach was originally designed to test two hypotheses under a sequence of identical and equally informative observations [15]. Extensions exist for the multi-hypothesis case [1]. However, stopping boundaries for the SPRT are not easy to derive when the observations are not equally informative. We describe an alternative approach based on concentration inequalities [4]. Unlike the SPRT, this approach is straightforward to apply even when there are multiple hypotheses and the observations are not equally informative. We have noted related ideas in other fields [11]; however, we believe our approach is novel in this context.

The techniques described in this paper were first presented in [6]. Here, we expand on that presentation with a more detailed justification of correctness and a more complete outline of how the techniques can be applied.

III. Probabilistic Bail Out Using Bennett's Inequality

Let $\mathcal{H} = \left\{H^1, \ldots, H^K\right\}$ be a set of K hypotheses (models), and let $Z = \left\{z_1, \ldots, z_N\right\}$ be an observation consists of N features. The likelihood of the observation under the kth hypothesis

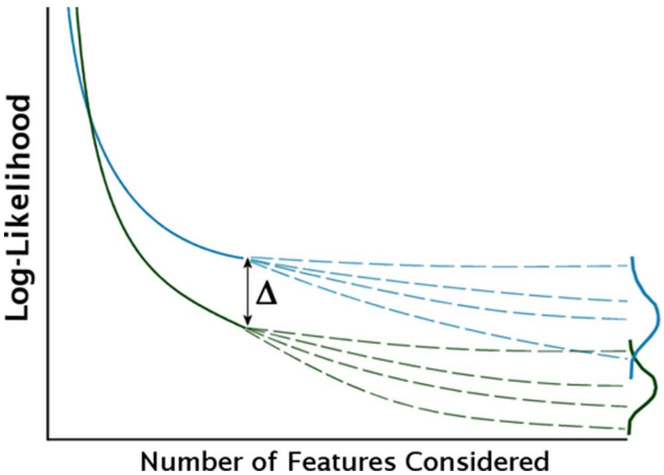

Fig. 1. Conceptual illustration of the bail-out test. After considering the first i features, the difference in log-likelihoods between two hypotheses is Δ. Given some statistics about the remaining features, it is possible to compute a bound on the probability that the evaluation of the remaining features will cause one hypothesis to overtake the other. If this probability is sufficiently small, the trailing hypothesis can be discarded.

is given by

$$p(Z|H^k) = p(z_1|z_2,\ldots,z_N,H^k)\ldots p(z_{N-1}|z_N,H^k) \times p(z_N|H^k). \quad (1)$$

Define the log-likelihood of the first i features under the kth hypothesis as follows:

$$D_i^k = \sum_{j=1}^{i} d_j^k \quad (2)$$

where

$$d_i^k = \ln(p(z_i|z_{i+1},\ldots,z_N,H_k)) \quad (3)$$

is the log-likelihood of the ith feature under the kth hypothesis. We would like to determine, as rapidly as possible, the hypothesis H^* for which the total log-likelihood D_N^* is maximized. To find H^* with certainty requires a complete evaluation of the likelihood of each hypothesis, which may be too slow for applications of interest. Consequently, we consider the problem of finding a hypothesis $H^{\#}$, subject to the constraint that $p(H^{\#} \neq H^*) < \epsilon$, where ϵ is some user-specified probability.

In overview, our approach is to calculate the likelihoods of all hypotheses in parallel and terminate the likelihood calculation for hypotheses that have fallen too far behind the current-leading hypothesis. "Too far" can be quantified using concentration inequalities, which yield a bound on the probability that a hypothesis will overtake the leader, given their current difference in likelihoods and some statistics about the properties of the features which remain to be evaluated. Fig. 1 illustrates the idea.

Consider two hypotheses $H^x, H^y \in \mathcal{H}$, and let

$$X_i = d_i^x - d_i^y \quad (4)$$

that is, the difference in the log-likelihood of feature i under hypothesis H^x and H^y. X_i can be considered to be a random variable before its value has been calculated. This is useful because we can calculate some key statistics about X_i more cheaply than we can determine its exact value. Now, let us define

$$S_n = \sum_{i=n+1}^{N} X_i. \quad (5)$$

If, after evaluating n features, the log-likelihood of some hypothesis is Δ less than the current best hypothesis, then the probability of failing to locate H^* if we discard this hypothesis is given by $p(S_n > \Delta)$, i.e., by the probability that the trailing hypothesis will overtake the leader after the evaluation of the remaining features. Thus, knowing the distribution of S_n enables the creation of a probabilistic bail-out test for discarding hypotheses subject to an error constraint. To calculate an explicit distribution on S_n is infeasible; however, concentration inequalities—which bound the probability that a function of random variables will deviate from its mean value—can be applied to yield bounds on $p(S_n > \Delta)$.

A large variety of concentration inequalities exist, many of which apply under very general conditions, including cases where the component distributions are not identically distributed, not independent, and combined using arbitrary functions. The more information available about the component distributions X_i, the tighter the bound. For an overview, see [4]. For our application, we will find it convenient to use the Bennett inequality for sums of symmetric random variables [3]. This inequality is specified in terms of two parameters—M, which is a bound on the maximum value of any component X_i, and v, which is a bound on the sum of the variances of the components X_i.

Formally, let $\{X_i\}_{i=n+1}^{N}$ be a collection of independent mean-zero random variables with symmetric distributions (corresponding to the log-likelihood changes due to those features not yet considered) and satisfying the conditions

$$p(|X_i| < M) = 1 \, \forall i : \{n+1 < i < N\} \quad (6)$$

$$\sum_{i=n+1}^{N} E\left[X_i^2\right] < v. \quad (7)$$

As before

$$S_n = \sum_{i=n+1}^{N} X_i. \quad (8)$$

Then, the Bennett inequality states that

$$p(S_n > \Delta) < \exp\left(\frac{v}{M^2}\cosh(f(\Delta)) - 1 - \frac{\Delta M}{v} f(\Delta)\right) \quad (9)$$

where

$$f(\Delta) = \sinh^{-1}\left(\frac{\Delta M}{v}\right). \quad (10)$$

Thus, after considering the first n features, and given bounds on the maximum value (M) and sum of variances (v) of the inter-hypothesis log-likelihood changes (X_i) for the features yet to be

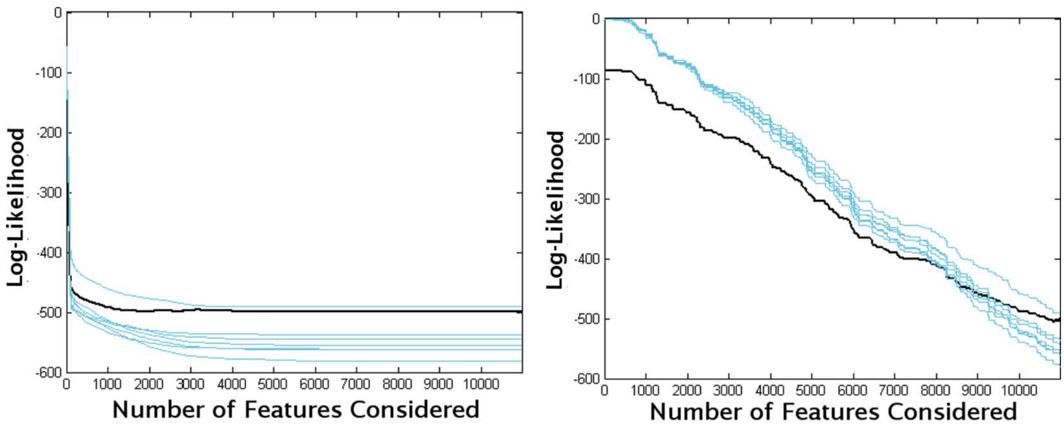

Fig. 2. Bail-out test on real data. Here, the blue lines show the log-likelihoods of each place versus number of features considered. Typically, there are thousands of places—here, only a few are shown for clarity. The black line is the bail-out threshold. Once the likelihood of a place hypothesis falls below the bail-out threshold, its likelihood calculation can be terminated (the remainder of the likelihood calculation is shown earlier for illustration). The calculations shown in (a) and (b) are the same, the only difference being the order in which features are considered. In (a), observations are ordered by information gain; in (b), they are ordered randomly. Note that ordering the features by information gain results in much faster convergence toward final likelihood values and, hence, a much more effective bail-out test. The bail-out threshold does not fully converge to the leading hypothesis because of the offset constant C described in Section V-D. (a) Features ordered by information gain. (b) Random feature order.

evaluated $(n+1:N)$, we can solve the Bennett inequality for Δ, such that $p(S_n > \Delta)$ meets the user-specified error probability ϵ. After considering a given feature, any hypothesis whose log-likelihood is at least Δ less than the current-leading hypothesis can be discarded, because the probability of it overtaking the current leader before the end of the calculation is less than ϵ. Note that the values of M and v are defined on the features that have yet to be evaluated $(n+1:N)$; thus, as the calculation progresses and the number of unconsidered features decreases, M and v will decrease. The bail-out threshold thus converges to zero as the number of unconsidered features decreases (see Fig. 2).

To apply this scheme to a concrete multi-hypothesis testing task, some method must be available to calculate the values M and v. In Section V, we will outline exactly how this can be achieved for our FAB-MAP place-recognition system. Note that if a particular problem does not allow for the calculation of these values, it might still be possible to apply the scheme by substituting a different concentration inequality. For example, the Hoeffding inequality [9] can be used when no information about v is available.

Before moving on, there are two implementation details worth mentioning. The first is how to solve the Bennet inequality for Δ. We do this using standard numerical techniques. To find Δ, we begin with a few iterations of the bisection method and then switch to Newton–Raphson iteration for faster convergence. If bisection has not brought us sufficiently close to the minima, then Netwon–Raphson may diverge. If we detect this behavior, we fall back to bisection for a few iterations. We also use the fact that the value of Δ after considering the first n features is strictly less than (and typically very close to) the value after considering $n-1$ features. Thus, after Δ has been found initially, it can be incrementally updated with only a single Netwon–Raphson iteration in most cases.

The second detail relates a low-level optimization of the log-likelihood calculation. We use the same trick as described by Nister in [13] to avoid unnecessary computation of the expensive *ln*-function. Instead of computing the log-likelihood increment for each feature, we take features in groups of ten and compute the log-likelihood increment for the whole group

$$\ln\left(\prod_{i=1}^{10} p(z_i|z_{i+1},\ldots,z_N,H_k)\right).$$

Grouping the features like this introduces a tradeoff, in that the bail-out test can now only be applied at every tenth step. The low-level computational gains must thus be balanced against the loss from less frequent bail-out tests. We experimented with different group sizes and found that ten gave the best performance. Groups much larger than ten should be avoided for numerical precision reasons.

IV. Appearance-Only Simultaneous Localization and Mapping

In the next section, we will describe the use of the bail-out technique with our FAB-MAP appearance-only SLAM system. However, we must first provide a brief description of how this system operates. FAB-MAP has been described in detail in [5] and [7], and a modified, more scalable version was described in [8]. We will present only a very brief overview here.

The design goal of the FAB-MAP system is to create a kind of appearance-based analog to metric SLAM. Whereas typical SLAM algorithms attempt to keep track of the pose of the robot in precise metric coordinates, FAB-MAP makes no attempt to track the vehicle. Instead, it simply classifies the current robot observation as belong to either a new never-before-seen location or one of the locations previously visited. If the location is new, it is added to the map so that it can be recognized in the future. Distinctive locations can be recognized even after unknown vehicle motion, thus making FAB-MAP suitable to solve problems such as a kidnapped robot, loop-closure detection, and multisession mapping, which are typically very hard to deal with in a purely metric SLAM framework. A notable aspect of FAB-MAP is that it is based on a principled probabilistic model, which means that the system naturally handles challenging cases, such as self-similar environments (*"perceptual aliasing"*), and makes full use of available information, including *negative observation* (discussed later) and correlations among observations.

Formally, at time t, the robot's map consists of n_t discrete locations, each location L_i having an associated appearance model. When the robot collects a new observation Z_t, we compute $p(L|Z_t)$, and the probability distribution over locations in the map give the current observation. This can be cast as a recursive Bayes filtering problem

$$p(L_i|\mathcal{Z}^t) = \frac{p(Z_t|L_i, \mathcal{Z}^{t-1})p(L_i|\mathcal{Z}^{t-1})}{p(Z_t|\mathcal{Z}^{t-1})} \tag{11}$$

where $\mathcal{Z}^t$ is the set of all observations up to time t, $p(Z_t|L_i, \mathcal{Z}^{t-1})$ is the likelihood of the observation given the location L_i and the previous observations $\mathcal{Z}^{t-1}$, $p(L_i|\mathcal{Z}^{t-1})$ is our prior belief about our location, and $p(Z_t|\mathcal{Z}^{t-1})$ normalizes the distribution.

To represent appearance, we use the *visual words* approach developed for image retrieval in the computer vision community [14]. Invariant features are detected in the current image (we use the speeded up robust features (SURF) detector [2]), and then, these features are quantized with respect to a *vocabulary* of prototypical features (learned from generic training data). An observation Z_t is then simply a binary vector, the kth entry of which indicates whether or not the kth word of the visual vocabulary was detected in the current image. A place appearance model L_i is similarly a vector of continuous probabilities and indicates our belief about the existence of visual-word-generating elements at that particular location.

A description of the FAB-MAP model is not the purpose of this paper. For detailed information on how the model is evaluated, see our earlier publications. For present purposes, it is sufficient to say that the primary cost for the evaluation of the FAB-MAP model is the calculation of the observation likelihood term $p(Z_t|L_i, \mathcal{Z}^{t-1})$, which must be evaluated for every location in the map every time the robot collects a new observation. The evaluation of this term can be expanded as follows:

$$p(Z_k|L_i) \approx p(z_r|L_i) \prod_{q=2}^{|v|} p(z_q|z_{p_q}, L_i) \tag{12}$$

where the terms z_q are the individual binary components of the observation Z_k, which indicates the presence or absence of a particular word in the visual vocabulary. The conditioning on z_{p_q} is because we account for some of the correlations between visual words. For details, see [7]. Our intention here is only to show that for each new observation, we have a set of hypotheses (the locations L_i), which we are evaluating under a set of features (the visual words z_q). Most locations will yield an insignificant likelihood of having generated the current observation. By identifying these locations before the calculation is fully complete, many locations can be quickly excluded from processing, and large speed increases can be realized.

V. Application of the Bail-Out Test to Appearance-Only SLAM

We now describe how to apply the bail-out test from Section III to our FAB-MAP model. To recap, we have a set of locations L_i whose likelihood we are evaluating with respect to some observation Z. The observation Z consists of a set of features z_i, each of which is a binary variable that indicates whether or not the ith word of the visual vocabulary is present in the observation. We will denote the difference in log-likelihood of two locations L_x and L_y under feature z_i by

$$\begin{aligned} X_i &= d_i^x - d_i^y \\ &= \ln(p(z_i|z_{p_i}, L_x)) - \ln(p(z_i|z_{p_i}, L_y)) \end{aligned} \tag{13}$$

which we will consider as a random variable before its value has been calculated. To apply our bail-out test, we must establish that our random variables X_i meet the conditions for applicability of Bennett's inequality. We must also define an order in which to consider the features and outline how M and v may be calculated.

A. Applicability of Bennett's Inequality

To make use of (9), we require that the random variables $\{X_i\}_{i=1}^N$ be independent and mean-zero and have symmetric distributions. We also require that their maximum values be bounded.

To understand why these conditions are satisfied, we note that, given that the values of z_i and z_{p_i} are fixed, the likelihood of a feature being observed at a particular location $p(z_i|z_{p_i}, L_x)$ depends only on the number of previous observations of feature z_i at location L_x (for a thorough appreciation of this point, it is necessary to refer the reader to [7]. Intuitively, however, this is because our belief about how likely the feature is to be observed at a location is initialized to some fixed prior, which is then only updated each time the feature is observed to be present or absent at the location). Taking this fact, and assuming the number of observations relating to any given place is finite, X_i has a multinomial distribution because $p(z_i|z_{p_i}, L_x)$ can be one of a fixed set of discrete values. If we now make the assumption that, *a priori*, all locations are equally likely to generate any given feature, X_i must have a symmetric mean-zero distribution. This is because if $X_i = \delta$ for some history of previous observations at L_x and L_y, then $X_i = -\delta$ for the symmetric case, where the observation history of L_x and L_y is swapped, which occurs with equal probability.

To illustrate this, consider an example. Let z_i be a feature, which is observed with probability η, and let L_x and L_y be two location models, which each have exactly one associated observation. Further assume that the state of the feature z_i in the observation associated with L_x and L_y (i.e., $z_i = 0$ or 1) is unknown. For compactness, denote the event that location L_x has a previous associated observation with $z_i = 0$ by $L_x\{0\}$. There are four possible cases, as shown in the following:

Case	Probability	Value of X_i
$L_x\{0\}, L_y\{0\}$	$p = (1-\eta)^2$	0
$L_x\{0\}, L_y\{1\}$	$p = \eta(1-\eta)$	δ
$L_x\{1\}, L_y\{0\}$	$p = \eta(1-\eta)$	$-\delta$
$L_x\{1\}, L_y\{1\}$	$p = \eta^2$	0

which occur with the probabilities shown. In addition, the log-likelihood difference (X_i) is shown in the four cases, where δ is the value of $\ln(p(z_i|z_{p_i}, L_x\{0\})) - \ln(p(z_i|z_{p_i}, L_y\{1\}))$. Observe that X_i has a symmetric, mean-zero multinomial distribution

$$X_i : \quad \begin{cases} \delta \text{ with probability } \eta(1-\eta) \\ 0 \text{ with probability } (1-\eta)^2 + \eta^2 \\ -\delta \text{ with probability } \eta(1-\eta). \end{cases} \qquad (14)$$

When the location models have more than one associated observation, this multinomial distribution has a similar form but a larger number of states.

Finally, Bennett's inequality requires that the variables X_i be independent. FAB-MAP incorporates a Chow Liu tree, which captures conditional dependencies between features. While incorporating this model ameliorates concerns about correlations, the Chow Liu tree is still only an approximation to the true joint distribution, and thus, the variables X_i may have weak residual dependence. However, our experimental results (see Section VI) would appear to indicate that this is not a problem in practice.

B. Ranking Features

Next, we must define an order in which to consider the features during the bail-out test. While the bail-out test applies to any ordering, it is natural to rank the features by information gain. This way, the hypotheses will converge most rapidly toward their final log-likelihood values, and poor hypotheses can be identified earliest (see Fig. 2).

Under our observation model, each feature z_i is conditionally dependent on one other feature z_{p_i}. If we observe $z_i = s_i$ and $z_{p_i} = s_{p_i}$ (with $s \in \{0, 1\}$), then the information gain associated with this observation under our model is as follows:

$$I = -\ln p(z_i = s_i | z_{p_i} = s_{p_i}). \qquad (15)$$

Typically observations of rare words are the highest ranked features, although, perhaps surprisingly, failure to observe a word can sometimes also have high information gain, for example, if two words are almost always observed together, then failure to observe one while observing the other is an informative observation.

Note that in our implementation, the probabilities in (15) come from the training data on which we learned the model of our visual words. As such, we are calculating the information gain with respect to the places in the training data. Strictly, we should consider the information gain with respect to the set of places in our current map, for example, some feature might be very rare in the training set but very common in the map. In practice, we think that maintaining a separate set of probabilities will usually be unnecessary. As the bail-out test applies to any feature ordering, the effect of a simplified implementation is merely that hypotheses may be discarded slightly later than optimal.

C. Calculating M and v

Finally, to apply Bennett's inequality, we must calculate M and v, i.e., the parameters of the concentration inequality, which depend on the component random variables X_i. As per (6), M is a bound on the maximum value of $X_i \forall i : \{n+1 < i < N\}$, i.e., the maximum of the log-likelihood differences due to feature i between some trailing location hypothesis L_x and the leading hypothesis at the time when feature i is considered, over all indices i between $n+1$ and N. The calculation of an exact value for M would require knowledge of the leading hypothesis at a future point in the calculation, which is unknown. However, as M is a *bound*on the maximum value of X_i, it can be calculated as the maximum interhypothesis log-likelihood change over *all* hypothesis pairs and all remaining features. This bound can easily be calculated by keeping track, for each feature i, of the location that was most and least likely to have generated that feature, as these location pairs maximize X_i.

The value of v can be determined directly as the sum of the variances of the distributions shown in (14). This is practical when the number of observations associated with each location in the map is small. However, as exploration continues and the number of observations associated with a typical place in the map increases, the calculation of this variance rapidly becomes expensive. At some point, it may be beneficial to switch from using Bennett's inequality to Hoeffding's inequality [9], which is a similar concentration inequality that requires knowledge only of the maximum value of each X_i. Hoeffding's inequality gives a weaker bound, but this is compensated for by the fact that by the time the variance becomes expensive to compute, the place models themselves are more differentiated, and therefore, their likelihoods will diverge faster. However, we have not yet investigated this issue because in the datasets we have labeled for evaluation, the robot typically visits a particular location only a small number of times.

D. Calculating a PDF Over Hypotheses

One remaining issue is that our appearance-only SLAM system requires a pdf over hypotheses, whereas our discussion so far has concerned locating only the best hypothesis. To compute a pdf requires a simple modification to the bail-out scheme. Consider that instead of locating only the best hypothesis H^*, we would like to locate all hypotheses, whose log-likelihood is at most C less than that of H^*. C is a user-specified constant chosen so that hypotheses less likely than this can be considered to have zero probability with minimal error. Simply increasing our bail-out distance by C will retain all those hypotheses whose final likelihood may be within this likelihood range, thus giving us a close approximation to the pdf over hypotheses.

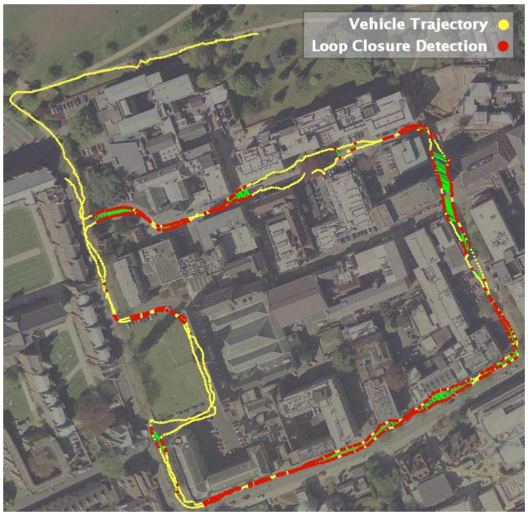

Fig. 3. Appearance-only matching results (using the accelerated algorithm) for the City Center dataset overlaid on an aerial photograph. The robot travels twice around a loop with total path length 2 km, collecting 2474 images. Each of these images is determined to be either a new place or a loop closure. Positions (from hand-corrected GPS) at which the robot collected an image are marked with a yellow dot. Two images that were assigned a probability $p \geq 0.99$ of having come from the same location are marked in red and joined with a green line. There are no incorrect matches that meet this probability threshold.

VI. Results

We tested the technique on FAB-MAP applied to imagery collected by a mobile robot. We use the New College and City Center datasets introduced in [7] and available online.[1] The ground truth for these data sets was labeled by hand. The binary for the FAB-MAP system we used is also available.[2] Regarding the datasets, the New College set has a trajectory of 1.9 km in length and features a number of challenging cases of perceptual aliasing. The City Center dataset is 2 km in length and is particularly rich in dynamic objects. We used a third dataset Parks Road, which features a typical suburban environment. In all three datasets, the robot collected images to the left and right of its trajectory approximately every 1.5 m. The robot's appearance model was built from a fourth dataset collected in a different region of the city, the area of which did not overlap with the test sets.

Navigation results for these datasets were generated using both the original system outlined in [7] and the accelerated system incorporating the bail-out test as described in this paper. All datasets were processed using the same visual vocabulary and algorithm parameters. The bail-out boundary was set so that the probability of incorrectly discarding the best hypothesis at any step was $<10^{-6}$. This value can be varied to trade off speed against accuracy, and we selected this value empirically as the best compromise.

Results are summarized in following the figures. Fig. 2 illustrates the bail-out calculation on some real data. Ordering the features by information gain clearly has a dramatic effect on the effectiveness of the bail-out test. Fig. 3 visualizes the overall performance of the accelerated algorithm on the City Center dataset. The system correctly identifies a large proportion of possible loop closures with high confidence. There are no false positives that meet the probability threshold necessary to accept a loop closure.

Precision-recall curves for the full and accelerated algorithms on the City Center dataset are shown in Fig. 4. The curves were

[1]http://www.robots.ox.ac.uk/∼mobile/IJRR_2008_Dataset/
[2]http://www.robots.ox.ac.uk/∼mjc/Software.htm

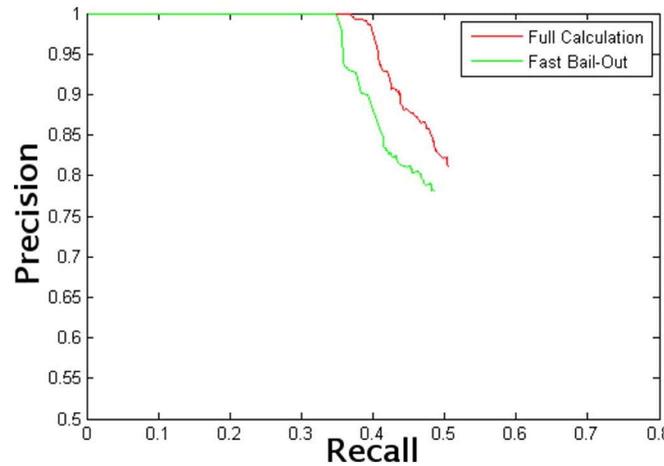

Fig. 4. Precision–recall curves for the City Center dataset show the full-likelihood calculation (red) and the accelerated calculation using the bail-out test (green). Note the offset on the axes.

TABLE I
COMPARISON OF THE PERFORMANCE OF THE SLAM SYSTEM USING FULL- AND ACCELERATED-LIKELIHOOD CALCULATIONS

	Full Calculation		Fast Bail-Out		
Dataset	Recall	Mean Time	Recall	Mean Time	Speed-Up
City Centre	37%	5015 ms	35%	141 ms	35.5
New College	46%	4818 ms	42%	178 ms	27.0
Parks Road	44%	4267 ms	40%	79 ms	53.6

Recall rates quoted are at 100% precision. Timing results are for the filter update, on a 3 GHz Pentium IV. Feature generation adds an extra 330 ms on average. Update time for the accelerated calculation is data dependent and varies from observation to observation. Time quoted is the average over the dataset.

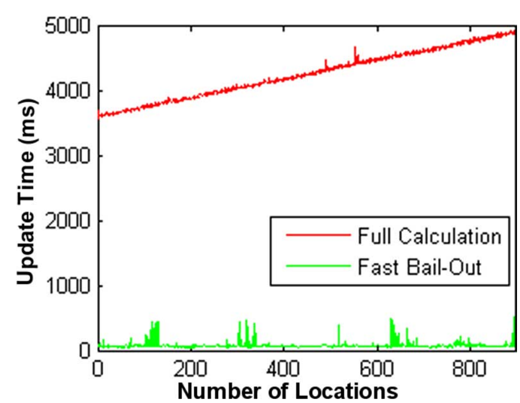

Fig. 5. Filter update time versus the number of locations in the map for the Parks Road dataset. Update time with zero locations is nonzero due to the fixed cost of evaluating the partition function in (11). Calculation time with the bail-out test grows linearly; however, the slope is too small to be seen on this graph.

(a) p=0.996 (b) p=0.999998 (c) p=0.999992

Fig. 6. Some examples of images that were assigned high probability of having come from the same place despite the scene change. Results were generated using the accelerated-likelihood calculation. Words common to both images are shown in green; others are shown in red. The probability that the two images come from the same place is indicated between the pairs.

generated by varying the probability at which a loop closure was accepted. Precision is defined as the ratio of true positive loop-closure detections to total detections. Recall is the ratio of true positive loop-closure detections to the number of ground truth loop closures. Note that images for which no loop closure exists cannot contribute to the true positive rate; however, they can generate false positives. Likewise true loop closures, which are incorrectly assigned to a "new place" depress recall but do not impact our precision metric. These metrics provide a good indication of how useful the system would be for loop-closure detection as part of a metric SLAM system—recall that 100% precision indicates the percentage of loop closures which can be detected with no false positives that would cause filter divergence. Note that recall rates are quoted in terms of image-to-image matches. As a typical loop closure is composed of multiple images, even a recall rate of 35% is sufficient to detect almost all loop closures. The relative performance of the two algorithms on the other datasets is summarized in Table I. Timing performance for the Parks Road dataset is shown in Fig. 5.

The speedup achieved by bail-out test exhibits a fairly broad range across the three datasets (from 27× to 53.6×; see Table I). The effectiveness of the bail-out tests is data dependent; therefore, this variation is not surprising. In general, we expect a large performance gain where the data contains distinctive features, which rapidly separate the location hypotheses. In cases where the features are less distinctive, for example due to higher perceptual aliasing in the environment, or lower level effects such as increased image blur, we would expect the speedup to be lower. This intuition, in fact, matches with the results here, as there is a substantial section of high-perceptual aliasing in the New College dataset, which we would tentatively suggest as the reason for the lower speedup on that set. This adaptive behavior seems desirable: when the data is very distinctive, the computation terminates quickly, and when the data is ambiguous, the algorithm considers more features to try to resolve the ambiguity.

Finally, Figs. 6 and 7 show some examples of place recognition performance, highlighting the fact that matching ability in the presence of scene change and robustness to perceptual aliasing is not significantly compromised by the bail-out test. The robustness to perceptual aliasing is particularly noteworthy. Of course, had the examples shown in Fig. 7 been genuine loop closures, they might also have received low probability of having come from the same place. We would argue that this is correct behavior, modulo the fact that the probabilities in Fig. 7(a) and (b) seem too low. The very low probabilities in Fig. 7(a) and (b) are due to the fact that the best matches for the query images are found in the sampling set used to suppress perceptual aliasing, which captures almost all the probability mass. This is less likely in the case of a true, but ambiguous loop closure, particularly because in the case of a true loop closure, the ambiguity can be resolved by temporal information via the prior term in (11). For a complete appreciation of these issues, see [5] and [7].

VII. Applicability to Other Robotics Problems

There many problems within robotics and computer vision which consist of ranking hypotheses using a set of features. The acceleration scheme discussed in this paper is very generic, and it seems that it could be applied to many of these other problems. Obvious candidates include object detection and localization, but it is likely other problems are suitable as well. Our method will be most useful when the set of hypotheses and features are both large, and it is feasible to quickly sort the

(a) $p=4.6\times 10^{-10}$ (b) $p=3\times 10^{-9}$ (c) $p=0.71$

Fig. 7. Some examples of remarkably similar-looking images from different parts of the workspace that were correctly assigned low probability of having come from the same place. Results were generated using the accelerated-likelihood calculation. The examples represent typical system performance. Words present in both images are shown in green; others are in red. (Common words are shown in blue in (b) for better contrast). The probability that the two images come from the same place is indicated between the pairs.

features by expected information gain (as shown in Fig. 2, this is crucial to obtain a useful bound). Problem-specific reasoning will be required to determine how to compute M and v for a particular application, but it does not seem that there is anything unusual about FAB-MAP which makes computing these quantities especially easy in our case. In particular, Hoeffding's bound, which requires knowledge only of M, could be applied to any problem where a set of hypotheses are evaluated against a sequence of independent features, and the maximum score contribution of each feature is known. It seems that this should be a sufficiently generic requirement to allow at least the Hoeffding bound to be used to accelerate many other algorithms. It is quite straightforward to implement the bail-out test once a method to compute M is available.

VIII. Conclusion

This paper has presented a new approach to rapid multi-hypothesis testing using a probabilistic bail-out condition based on concentration inequalities. Concentration inequalities exist that apply under very general conditions, even for arbitrary functions of non-independent and non-identically distributed random variables; hence, our basic idea should be applicable to a wide variety of problems. The approach in effect generalizes techniques such as the SPRT, which have already shown great utility in applications such as efficient RANSAC. Unlike the SPRT, our technique is easy to apply even when observations are not equally informative and the hypothesis test is not binary. We show how to apply our bail-out test to accelerate the FAB-MAP appearance-only SLAM system. The speed increase obtained is data-dependent, but acceleration factors in the range $25\times$–$50\times$ are typical in our tests. The location-recognition performance of the accelerated system is only marginally less than that of the full solution and more than sufficient for reliable online loop-closure detection in mobile robotics applications. We have presented results demonstrating online loop-closure detection over 2 km loops, although the system is fast enough to scale to loops of tens of kilometers in length, while maintaining subsecond filter update times. While we have subsequently explored even more efficient computational schemes for the specific task tackled by FAB-MAP [5], [8], the acceleration technique presented in this paper is much more general in its applicability and should find uses in problems beyond appearance-based SLAM.

References

[1] C. W. Baum and V. Veeravalli, "A sequential procedure for multihypothesis testing," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1994–2007, Nov. 1994.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[3] G. Bennett, "Probability inequalities for the sum of independent random variables," *J. Amer. Statist. Assoc.*, vol. 57, pp. 33–45, 1962.

[4] S. Boucheron, G. Lugosi, and O. Bousquet, *Concentration Inequalities,* Lecture Notes in Artificial Intelligence, vol. 3176, New York: Springer-Verlag, 2004, pp. 208–240.

[5] M. Cummins, "Probabilistic localization and mapping in appearance space," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 2009.

[6] M. Cummins and P. Newman, "Accelerated appearance-only SLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, 2008, pp. 1828–1833.

[7] M. Cummins and P. Newman. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance.

Int. J. Robot. Res. [Online]. *27(6)*, pp. 647–665. Available: http://ijr.sagepub.com/cgi/content/abstract/27/6/647

[8] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM—FAB-MAP 2.0," in *Proc. Robot.: Sci. Syst.*, Seattle, WA, 2009.

[9] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963.

[10] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[11] O. Maron and A. W. Moore. (1994). "Hoeffding races: Accelerating model selection search for classification and function approximation," presented at the Adv. Neural Inf. Process. Syst., Vancouver, BC, Canada. [Online]. Available: citeseer.ist.psu.edu/maron94hoeffding.html

[12] J. Matas and O. Chum, "Randomized RANSAC with sequential probability ratio test," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1727–1732.

[13] D. Nistér, "Preemptive RANSAC for live structure and motion estimation," *Mach. Vis. Appl.*, vol. 16, no. 5, pp. 321–329, 2005.

[14] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," presented at the Int. Conf. Comput. Vis., Nice, France, 2003.

[15] A. Wald, *Sequential Analysis*. New York: Dover, 1947.

Authors' photographs and biographies not available at the time of publication.