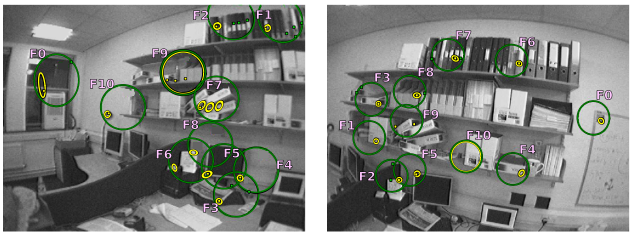
Efficient Data Association in Images using Active Matching

Margarita Chli and Andrew J. Davison

{mchli, ajd}@doc.ic.ac.uk
Department of Computing
Imperial College London
London SW7 2AZ, UK

***Abstract*— In the feature matching tasks which form an integral part of visual tracking or SLAM, there are invariably priors available on the absolute and/or relative image locations of features of interest. Usually, these priors are used post-hoc in the process of resolving feature matches and obtaining final scene estimates, via 'first get candidate matches, then resolve' consensus algorithms such as RANSAC or JCBB. In this paper we show that the dramatically different approach of using priors dynamically to guide a feature by feature matching search can achieve global matching with much fewer image processing operations and lower overall computational cost. Essentially, we put image processing *into the loop* of the search for global consensus. In particular, our approach is able to cope with significant image ambiguity thanks to a dynamic mixture of Gaussians treatment. In our fully Bayesian algorithm, the choice of the most efficient search action at each step is guided intuitively and rigorously by expected Shannon information gain. We demonstrate the algorithm in feature matching as part of a sequential SLAM system for 3D camera tracking. Robust, real-time matching can be achieved even in the previously unmanageable case of jerky, rapid motion necessitating weak motion modelling.**



(a) Fast camera motion at 15Hz (b) Slow camera motion at 15Hz

Fig. 1. Active matching dramatically reduces image processing operations while still achieving global matching consensus. Sequence results: superposition of the green individual gating ellipses searched in order to generate candidates for outlier rejection by JCBB and the yellow ellipses searched for our Active Matching [1] method. In these frames, joint compatibility needed to search a factor of 8.4 more image area than active matching in (a) and a factor or 4.8 in (b). JCBB must resolve all the matches shown (blobs), whereas Active Matching only finds the yellow blobs.

I. Introduction

It is well known that the key to obtaining correct feature associations in potentially ambiguous matching (data association) tasks is to search for a set of correspondences which are in *consensus*: they are all consistent with a believable global hypothesis. The usual approach taken to search for matching consensus is as follows: first candidate matches are generated, for instance by detecting all features in both images and pairing features which are nearby in image space and have similar appearance. Then, incorrect 'outlier' matches are pruned by proposing and testing hypotheses of global parameters which describe the world state of interest — the 3D position of an object or the camera itself, for instance. The sampling and voting algorithm RANSAC [6] has been widely used to achieve this in geometrical vision problems.

The idea that inevitable outlier matches must be 'rejected' from a large number of candidates achieved by some blanket initial image processing is deeply entrenched in computer vision and robotics. The approach of our *active matching* paradigm is very different — to cut outliers out at source wherever possible by searching only the parts of the image where true positive matches are most probable. Instead of searching for all features and then resolving, feature searches occur one by one. The results of each search, via an exhaustive but concentrated template checking scan within a region, affect the regions within which it is likely that each of the other features will lie. This is thanks to the same inter-feature correlations of which standard consensus algorithms take advantage — but our algorithm's dynamic updating of these regions within the matching search itself means that low probability parts of the image are *never examined at all* (see Figure 1), and the number of image processing operations required to achieve global matching is reduced by a large factor. Information theory intelligently guides the step by step search process from one search region to the next and can even indicate when matching should be terminated at a point of diminishing returns.

Davison [4] presented a theoretical analysis of information gain in sequential image search. However, Davison's work had the serious limitation of representing the current estimate of the state of the search at all times with a single multi-variate Gaussian distribution. This meant that while theoretically and intuitively satisfying active search procedures were demonstrated in simulated problems, the technique was not applicable to real image search because of the lack of ability to deal with discrete multiple hypotheses which arise due to matching ambiguity — only simulation results were given. Here we use a dynamic mixture of Gaussians (MOG) representation which grows as necessary to represent the discrete multiple hypotheses arising during active search. We show that this representation can now be applied to achieve highly efficient image search in real, ambiguous tracking problems.

Matching constraints are obtained by projecting an uncertain world state into a new image, the general result being a joint prior probability distribution over the image locations of features. These uncertain feature *predictions* will often be highly correlated. When probabilistic priors are available, the unsatisfying random sampling and preset thresholds of RANSAC have been improved on by probabilistic methods such as the Joint Compatibility Branch and Bound (JCBB) algorithm [10] which matches features via a deterministic interpretation tree [8] and has been applied to matching in visual SLAM [2]. JCBB takes account of a joint Gaussian prior on feature positions and calculates the joint probability that any particular hypothesized set of correspondences is correct. Our algorithm aims to perform at least as well as JCBB in determining global consensus while searching much smaller regions of an image. It has the additional advantage over JCBB of taking full account of probabilistic prior information which may be available on per-feature statistics such as false positive and false negative probabilities. These play a part in guiding search and are accounted for in the final probabilistic evaluation of hypotheses.

In this paper we introduce the active matching algorithm, to be motivated more fully in upcoming publication [1], and place emphasis on a clear statement of the algorithm and experimentation in a number of different feature matching situations.

II. Active Matching

In real image search problems no match (or failed match) can be fully trusted: true matches are sometimes missed (false negatives), and clutter similar in appearance to the feature of interest can lead to false positives. Modelling the probablilistic 'search state' as a mixture of Gaussians, we wish to retain the feature-by-feature quality of active search [4]. Our new MoG representation allows dynamic, online updating of the multi-peaked PDF over feature locations which represents the multiple hypotheses which arise during as features are matched ambiguously.

Our active matching algorithm searches for global correspondence in a series of steps which gradually refine the probabilistic search state initially set as the prior on feature positions. Each step consists of a search for a template match to one feature within a certain bounded image region, followed by an update of the search state which depends on the search outcome. After many well-chosen steps the search state collapses to a highly peaked posterior estimate of image feature locations — and matching is finished.

A. Search State Mixture of Gaussians Model

A single multi-variate Gaussian probability distribution over the vector $\mathbf{x}_m$ which stacks the object state and candidate measurements, is parameterised by a 'mean vector' $\hat{\mathbf{x}}_m$ and its full covariance matrix $\mathtt{P}_{\mathbf{x}_m}$. We use the shorthand $\mathbf{G}(\hat{\mathbf{x}}_m, \mathtt{P}_{\mathbf{x}_m})$ to represent the explicit normalised PDF:

$$p(\mathbf{x}_m) = \mathbf{G}(\hat{\mathbf{x}}_m, \mathtt{P}_{\mathbf{x}_m}) \qquad (1)$$

$$= (2\pi)^{-\frac{D}{2}} |\mathtt{P}_{\mathbf{x}_m}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_m - \hat{\mathbf{x}}_m)^\top \mathtt{P}_{\mathbf{x}_m}^{-1}(\mathbf{x}_m - \hat{\mathbf{x}}_m)} \ . \qquad (2)$$

During active matching, we now represent the PDF over $\mathbf{x}_m$ with a multi-variate MOG distribution formed by the sum of K individual Gaussians each with weight λ_i:

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}_i) = \sum_{i=1}^{K} \lambda_i \mathbf{G}_i \ , \qquad (3)$$

where we have now used the further notational shorthand $\mathbf{G}_i = \mathbf{G}(\hat{\mathbf{x}}_{m_i}, \mathtt{P}_{\mathbf{x}_{m_i}})$. Each Gaussian distribution must have the same dimensionality and the weights must normalise $\sum_{i=1}^{K} \lambda_i = 1$ for this to be a valid PDF.

The current MOG search state model forms the prior for a step of active matching. This prior, and the likelihood and posterior distributions to be explained in the following sections, are shown in symbolic 1D form in Section II-D.

B. The Algorithm

The MoG active matching process is initialized with a joint Gaussian prior over the features' locations in measurement space (e.g. prediction after application of motion model). Hence, at start-up the mixture consists of a single multivariate Gaussian. The process of selecting the {Gaussian, Feature} measurement pair to measure in the next matching step involves assessing the amount of information gain that each candidate pair is expected to provide. This is explained in detail in Section III.

```
ActiveMatching($\mathbf{G}_0$)

Mixture = $[1, \mathbf{G}_0]$
[f$_p$,$\mathbf{G}_p$, max-gain] = max_gain_candidate(Mixture)
**while** (not_yet_measured(f$_p$, $\mathbf{G}_p$) & max-gain $> I_{max}$)
    Matches = measure(f$_p$, $\mathbf{G}_p$)
    UpdateMixture(Mixture, $p$, f$_p$, Matches)
    prune_insignificant_gaussians(Mixture)
    [f$_p$,$\mathbf{G}_p$, max-gain] = max_gain_candidate(Mixture)
**end while**
$\mathbf{G}_{best}$ = find_most_probable_gaussian(Mixture)
**return** $\mathbf{G}_{best}$
```

For every template match yielding by the search of the selected {Gaussian, Feature} measurement pair a new Gaussian is spawned with mean and covariance conditioned on the hypothesis of that match being a true positive — this will be a more peaked than its parent. In both cases of either a successful or null template search the weights of the existing Gaussians are redistributed to reflect the current MoG search state. The full description of the update step after a measurement is detailed in the rest of this section.

Finally, very weak Gaussians (with weight < 0.001) are pruned from the mixture after each search step. This avoids otherwise rapid growth in the number of Gaussians such that in practical cases fewer than 10 Gaussians are 'live' at any point, and most of the time much fewer than this. This pruning is the better, fully probabilistic equivalent in the dynamic MOG scheme of lopping off branches in an explicit interpretation tree search such as JCBB [10].

```
UpdateMixture(Mixture$_{1..K}$, $i$, f, Matches$_{1..M}$)
```

Propagate the result of measuring f in $\mathbf{G}_i$ in the Mixture

```
[$\lambda_i, \mathbf{G}_i$] = Mixture[$i$]
for $k = 1 : K$
    if $k = i$ then
        for $m = 1 : M$
            $\mathbf{G}_m$ = fuse_match($\mathbf{G}_i$, Matches[$m$])
            $\lambda_m$ = $\lambda_i \times$scaling_factor($\mathbf{G}_m$, $\mathbf{G}_i$, Matches[$m$], f)
            Mixture[$i$] = [$\lambda_m, \mathbf{G}_m$]
        end for
        $\lambda_i = \lambda_i \times$scaling_factor($\mathbf{G}_i$, $\mathbf{G}_i$, Matches, f)
        Mixture[$i$] = [$\lambda_i, \mathbf{G}_i$]
    else
        $\lambda_k = \lambda_k \times$scaling_factor($\mathbf{G}_k$, $\mathbf{G}_i$, Matches, f)
        Mixture[$k$] = [$\lambda_k, \mathbf{G}_k$]
    end if
end for
normalize_weights(Mixture)
return
```

Note: scaling_factor($\mathbf{G}_k$, $\mathbf{G}_i$, Matches, f) *returns the factor by which the weight of $\mathbf{G}_k$ should be scaled given that feature* f *has been measured in $\mathbf{G}_i$ to produce the template matches in* Matches. *This result obeys the top line of Equation 9.*

C. Likelihood Function

One step of active matching which takes place by searching the region defined by the high-probability 3σ extent of one of the Gaussians in the measurement space of the selected feature. If we find M candidate template matches and no match elsewhere $\mathbf{z}_c = \{\mathbf{z}_1 \ldots \mathbf{z}_M \mathbf{z}'_{rest}\}$ then the likelihood $p(\mathbf{z}_c|\mathbf{x})$ of this result is modelled as a mixture: M Gaussians $\mathbf{H}_m$ representing the hypotheses that each candidate is the true **match** (these Gaussians, functions of $\mathbf{x}$, having the width of the measurement uncertainty $\mathtt{R}_i$), and two constant terms representing the hypotheses that the candidates are all spurious false positives and the true match lies either **in** or **out** of the searched region:

$$p(\mathbf{z}_c|\mathbf{x}) = \mu_{\text{in}}\mathbf{T}_{\text{in}} + \mu_{\text{out}}\mathbf{T}_{\text{out}} + \sum_{m=1}^{M} \mu_{\text{match}}\mathbf{H}_m \ . \qquad (4)$$

If N is the total number of pixels in the search region, then the constants in this expression have the form:

$$\begin{aligned} \mu_{\text{in}} &= P_{\mathbf{fp}}^{M} P_{\mathbf{fn}} P_{\mathbf{tn}}^{N-(M+1)} && (5) \\ \mu_{\text{out}} &= P_{\mathbf{fp}}^{M} P_{\mathbf{tn}}^{N-M} && (6) \\ \mu_{\text{match}} &= P_{\mathbf{tp}} P_{\mathbf{fp}}^{M-1} P_{\mathbf{tn}}^{N-M} \ , && (7) \end{aligned}$$

where $P_{\mathbf{tp}}$, $P_{\mathbf{fp}}$ $P_{\mathbf{tn}}$, $P_{\mathbf{fn}}$ are true positive, false positive, true negative and false negative probabilities respectively for the feature. $\mathbf{T}_{\text{in}}$ and $\mathbf{T}_{\text{out}}$ are top-hat functions with value one inside and outside of the searched Gaussian $\mathbf{H}_m$ respectively and zero elsewhere, since the probability of a null search depends on whether the feature is really within the search region or not. Given that there can only be one true match in the searched region, μ_{in} represents the probability that we record M false positives, one false negative and $N-(M+1)$ true negatives. μ_{out} represents the probability of M false positives and $N-M$ true negatives. The μ_{match} weight of a Gaussian hypothesis of a true match represents one true positive, $M-1$ false positives and $N-M$ true negatives.

D. Posterior: Updating After a Measurement

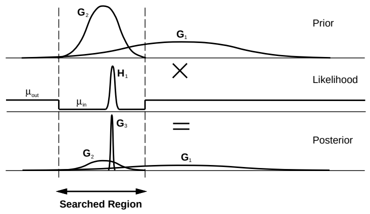
The standard application of Bayes' Rule to obtain the posterior distribution for $\mathbf{x}$ given the new measurement is:

$$p(\mathbf{x}|\mathbf{z}_c) = \frac{p(\mathbf{z}_c|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}_c)} \ . \qquad (8)$$

Substituting MOG models from Equations 3 and 4:

$$p(\mathbf{x}|\mathbf{z}_c) = \frac{\left(\mu_{\text{in}}\mathbf{T}_{\text{in}} + \mu_{\text{out}}\mathbf{T}_{\text{out}} + \sum_{m=1}^{M} \mu_{\text{match}}\mathbf{H}_m\right)\left(\sum_{i=1}^{K} \lambda_i \mathbf{G}_i\right)}{p(\mathbf{z}_c)} \ . \qquad (9)$$

The denominator $p(\mathbf{z}_c)$ is a constant determined by normalising all new weights λ_i to add up to one). In the illustration below illustrating the formation of a posterior, we show an example of $M = 1$ match. This posterior will then become the prior for the next active matching step.



In the top line of Equation 9, the product of the two MOG sums will lead to K scaled versions of all the original Gaussians and MK terms which are the products of two Gaussians. However, we make the approximation that only M of these MK Gaussian product terms are significant: those involving the prior Gaussian currently being measured. We assume that since the other Gaussians in the prior distribution are either widely separated or have very different weights, the resulting products will be negligible. Therefore there are only M new Gaussians added to the mixture: generally highly-weighted, spiked Gaussians corresponding to new matches in the searched region. These are considered to be '*children*' of the searched parent Gaussian. An important point to note is that if multiple matches in a search region lead to several new child Gaussians being added, one corresponding to a match close to the centre of the search region will correctly have a higher weight than others, having been formed by the product of a prior and a measurement Gaussian with nearby means.

All other existing Gaussians get updated posterior weights by multiplication with the constant terms. Note that the null information of making a search where no template match is found is fully accounted for in our framework — in this case we will have $M = 0$ and no new Gaussians will be generated, but the weight of the searched Gaussian will diminish.

III. Measurement Selection

.

We assume that the input prior at the start of the search process is well-represented by a single Gaussian and therefore

$\lambda_1 = 1$. As active search progresses and there is a need to propagate multiple hypotheses, this and subsequent Gaussians will divide as necessary, so that at a general instant there will be K Gaussians with normalised weights.

A. Search Candidates

At each step of the MOG active matching process, we use the mixture $p(\mathbf{x}_m)$ to predict individual feature measurements, and there are KF possible actions, where F is the number of measurable features. We rule out any $\{$Gaussian, Feature$\}$ combinations where we have already made a search. Also ruled out are 'child' Gaussians for a certain feature which lie completely within an already searched ellipse. For example, if we have measured root Gaussian $\mathbf{G}_1$ at feature 1, leading to the spawning of $\mathbf{G}_2$ which we search at feature 3 to spawn $\mathbf{G}_3$, then the candidates marked with '$*$' would be ruled out from selection:

	$\mathbf{F_1}$	$\mathbf{F_2}$	$\mathbf{F_3}$	$\mathbf{F_4}$	$\mathbf{F_5}$	$\mathbf{F_6}$
$\mathbf{G_1}$	$*$					
$\mathbf{G_2}$	$*$		$*$			
$\mathbf{G_3}$	$*$		$*$			

(10)

All of the remaining candidates are evaluated in terms of mutual information with the state or other candidate measurements, and then selected based on an information efficiency score [4] which is this mutual information divided by the area of the search region, assumed proportional to search cost.

B. Mutual Information for a Mixture of Gaussians Distribution

In order to assess the amount of information that each candidate $\{$Feature, Gaussian$\}$ measurement pair can provide, we predict the post-search mixture of Gaussians depending on the possible outcome of the measurement: (1): A **null search**, where no template match is found above a threshold. The effect is only to change the weights of the current Gaussians in the mixture into λ'_i. (2): A **template match**, causing a new Gaussian to be spawned with reduced width as well as re-distributing the weights of the all Gaussians of the new mixture to λ''_i.

In a well-justified assumption of 'weakly-interacting Gaussians' which are either well-separated or have dramatically different weights, we separate the information impact of each candidate measurement into two components: (a) I_{discrete} captures the effect of the redistribution of weights depending on the search outcome and (b) $I_{\text{continuous}}$ gives a measure of the reduction in the uncertainty in the system on a match-search. Due to the intuitive absolute nature of mutual information, these terms are additive:

$$I = I_{\text{discrete}} + I_{\text{continuous}} \tag{11}$$

One of other of these terms will dominate at different stages of the matching process, depending on whether the key uncertainty is due to discrete ambiguity or continuous accuracy. It is highly appealing that this behaviour arises automatically thanks to the MI formulation.

1) Mutual Information: Discrete Component: Considering the effect of a candidate measurement purely in terms of the change in the weights of the Gaussians in the mixture, we calculate the mutual information that is predicted to provide by

$$I(\mathbf{x};\mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}). \tag{12}$$

Given that the search outcome can have two possible states (null or match-search), then:

$$\begin{aligned} I_{\text{discrete}} = H(\mathbf{x}) \quad &- \quad P(\mathbf{z}=\text{null}) \quad H(\mathbf{x}|\mathbf{z}=\text{null}) & (13) \\ &- \quad P(\mathbf{z}=\text{match})H(\mathbf{x}|\mathbf{z}=\text{match}) \ . & (14) \end{aligned}$$

where

$$\begin{aligned} H(\mathbf{x}) &= \sum_{i=1}^{K} \lambda_i \log_2 \frac{1}{\lambda_i} & (15) \\ H(\mathbf{x}|\mathbf{z}=\text{null}) &= \sum_{i=1}^{K} \lambda'_i \log_2 \frac{1}{\lambda'_i} & (16) \\ H(\mathbf{x}|\mathbf{z}=\text{match}) &= \sum_{i=1}^{K+1} \lambda''_i \log_2 \frac{1}{\lambda''_i} \ . & (17) \end{aligned}$$

The predicted weights after a null or a match search are calculated as in Equation 9 with the only difference that the likelihood of a match-search is summed over all positions in the search-region that can possibly yield a match.

2) Mutual Information: Continuous Component: Davison [4], building on early work by others such as Manyika [9], explained clearly that the Mutual Information (MI) between a candidate and the scene state is the essential probabilistic measure of measurement value. With his single Gaussian formulation, has shown that the mutual information in bits between any two partitions α and β of $\mathbf{x}_m$ can be calculated according to this formula:

$$I(\alpha;\beta) = \frac{1}{2}\log_2 \frac{|\mathtt{P}_{\alpha\alpha}|}{|\mathtt{P}_{\alpha\alpha} - \mathtt{P}_{\alpha\beta}\mathtt{P}_{\beta\beta}^{-1}\mathtt{P}_{\beta\alpha}|} \ , \tag{18}$$

where $\mathtt{P}_{\alpha\alpha}$, $\mathtt{P}_{\alpha\beta}$, $\mathtt{P}_{\beta\beta}$ and $\mathtt{P}_{\beta\alpha}$ are sub-blocks of $\mathtt{P}_{\mathbf{x}_m}$. This representation however can be computationally expensive as it involves matrix inversion and multiplication so exploiting the properties of mutual information we can reformulate into:

$$\begin{aligned} I(\alpha;\beta) &= H(\alpha) - H(\alpha|\beta) & (19) \\ &= H(\alpha) + H(\beta) - H(\alpha,\beta) & (20) \\ &= \frac{1}{2}\log_2 \frac{|\mathtt{P}_{\alpha\alpha}||\mathtt{P}_{\beta\beta}|}{|\mathtt{P}_{\mathbf{x}_m}|} \ . & (21) \end{aligned}$$

Therefore we use $\mathtt{P}_{\alpha\alpha} = \mathtt{P}_{\mathbf{z}_{T \neq i}}$ and $\mathtt{P}_{\beta\beta} = \mathtt{P}_{\mathbf{z}_{T=i}}$ to compute the Continuous MI by

$$I_{\text{continuous}} = \frac{1}{2}P(\mathbf{z}=\text{match})\lambda''_m \log_2 \frac{|\mathtt{P}_{\alpha\alpha}||\mathtt{P}_{\beta\beta}|}{|\mathtt{P}_{\mathbf{x}_m}|} \tag{22}$$

This captures the information gain associated with the shrinkage of the measured Gaussian (λ''_m is the predicted weight of the new Gaussian evolving) thanks to the positive match: if the new Gaussian has half the determinant of the old one, that is one bit of information gain. This was the only MI term considered in [4] but is now scaled and combined with discrete component arising due to the expected change in the λ_i distribution.

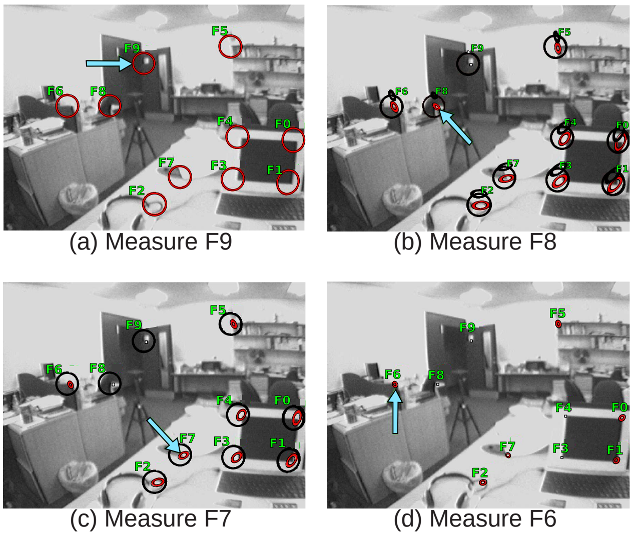
IV. Results

We present results on the application of the algorithm to feature matching for several different situations within the publically available MonoSLAM system [5] for real-time probabilistic structure and motion estimation — during normal tracking over long sequences, for high frame-rate tracking and during modest loop closure within single-map SLAM.

MonoSLAM, which is well known for its computational efficiency thanks to simple predictive search, uses an Extended Kalman Filter to estimate the joint distribution over the 3D location of a calibrated camera and a sparse set of point features — here we use it to track the motion of a hand-held camera in an office scene with image capture normally at 30Hz. At each image of the real-time sequence, MonoSLAM applies a probabilistic motion model to the accurate posterior estimate of the previous frame, adding uncertainty to the camera part of the state distribution. In standard configuration it then makes independent probabilistic predictions of the image location of each of the features of interest, and each feature is independently searched for by an exhaustive template matching search within the ellipse defined by a three standard deviation gate. The top-scoring template match is taken as correct if its normalised SSD score passes a threshold. At low levels of motion model uncertainty, mismatches via this method are relatively rare, but in advanced applications of the algorithm [2, 11] it has been observed that Joint Compatibility testing finds a significant number of matching errors and greatly improves performance.

Our active matching algorithm simply takes as input from MonoSLAM the predicted stacked measurement vector $\mathbf{z}_T$ and innovation covariance matrix $\mathtt{S}$ and returns a list of globally matched feature locations. We have implemented a straightforward feature statistics capability within MonoSLAM to sequentially record the average number of locations in an image similar to each of the mapped features, counting successful and failed match attempts in the feature's true location. This is used to assess false positive and false negative rates for each feature. More sophisticated online methods for assessing feature statistics during mapping have recently been published [3] and it would be intriguing to incorporate a bag of words feature model with active matching.

A. Sequence Results



(a) Measure F9 (b) Measure F8

(c) Measure F7 (d) Measure F6

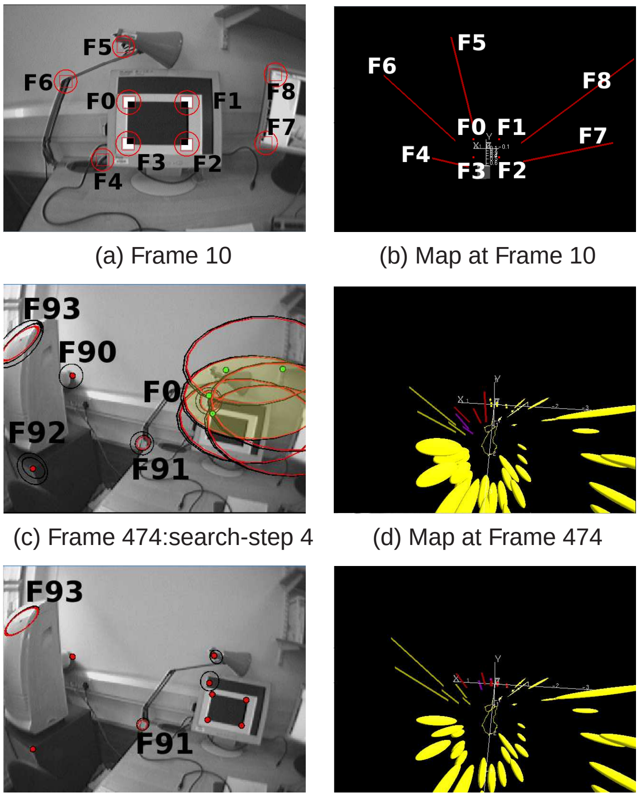
Fig. 2. Resolving ambiguity in MonoSLAM using active matching. Starting from (a) showing single Gaussian $\mathbf{G}_0$ set to the image prior at the start of matching, red ellipses represent the most probable Gaussian at each step and the arrows denote the {Feature,Gaussian} combination selected for measurement guided by MI efficiency. Feature 9 yields 2 matches and therefore two new Gaussians evolve in (b), $\mathbf{G}_1$ (small red) and $\mathbf{G}_2$(small black). Successful measurement of Feature 8 in $\mathbf{G}_1$ lowers the weight of $\mathbf{G}_2$ (0.00013) so in (c) it gets pruned from the mixture. Despite the unsuccessful measurement of Feature 7 in $\mathbf{G}_1$, after successful measurements of Features 3 and 4, there is only one Gaussian left in the mixture, with very small search-regions for all yet-unmeasured features.

Two different hand-held camera motions were used to capture image sequences at 30Hz: one with a standard level of dynamics slightly faster than of in the results of [5], and one with much faster, jerky motion. MonoSLAM's motion model parameters were tuned such that prediction search regions were wide enough that features did not 'jump out' at any point — necessitating a large process noise covariance and very large search regions for the fast sequence. Two more sequences were generated by subsampling each of the 30Hz sequences by a factor of two. These four sequences were all processed using active matching and also the combination of full searches of all ellipses standard in MonoSLAM with JCBB to prune outliers. Figure 2 shows an example of the sequential resolution of ambiguity carried out by active matching. In terms of accuracy, active matching was found to determine the same set of feature associations as JCBB on all frames of the sequences.

The key difference was in the computational requirements of the algorithms, as shown below:

	One tracking step	Matching only	No. pixels searched	Max no. live Gaussians
	Fast Sequence at 30Hz (752 frames)			
JCBB	56.8ms	51.2ms	40341	
Active Matching	21.6ms	16.1ms	5039	7
	Fast Sequence at 15Hz (376 frames)			
JCBB	102.6ms	97.1ms	78675	
Active Matching	38.1ms	30.4ms	9508	10
	Slow Sequence at 30Hz (592 frames)			
JCBB	34.9ms	28.7ms	21517	
Active Matching	19.5ms	16.1ms	3124	5
	Slow Sequence at 15Hz (296 frames)			
JCBB	59.4ms	52.4ms	40548	
Active Matching	22.0ms	15.6ms	5212	6

The key result here is the ability of active matching to cope efficiently with global consensus matching at real-time speeds (looking at the 'One tracking step' total processing time column in the table) even for the very jerky camera motion which is beyond the real-time capability of the standard 'search all ellipses and resolve with JCBB' approach whose processing times exceed real-time constraints. This computational gain is due to the large reductions in the average number of template matching operations per frame carried out during feature search, as highlighted in the 'No. pixels searched' column — Global consensus matching has been achieved by analysing around one eighth of the image locations needed by standard techniques. (JCBB itself, given match candidates, runs typically in 1ms per frame.) This is



(a) Frame 10
(b) Map at Frame 10
(c) Frame 474:search-step 4
(d) Map at Frame 474
(e) Frame 474: complete
(f) Map at Frame 475

Fig. 3. Loop Closing using Active Matching. The measurement of aged features has been delayed until at least 4 of them can be visible at the same frame. (a) shows a frame where the first features in the map have been initialized, and (b) shows the state of the map at that instant. Traversing a loop-like trajectory, the camera returns at a previously visited scene at (c) and (d) where features $\{F0, ..., F5\}$ are predicted to be visible with the relevant uncertainties projected as 2D ellipses in (c). Active matching attempts to localize the most certain features first, and then goes on to measure $F0$ in the highlighted ellipse in (c), yielding 4 matches for that feature. The ambiguity is then resolved by checking for consensus in the rest of the features, and rejecting all false positives to complete the matching in (e). (f) shows the map after propagating the loop-closure feature associations to correct both the estimated camera trajectory and reduce the uncertainty in the map.

illustrated dramatically in Figure 1, where the regions of pixels actually searched by the two techniques are overlaid on frames from two of the sequences.

B. Loop Closing using Active Matching

To investigate the use of active matching in a situation with even weaker prior, we present results of modest loop closing within a single EKF MonoSLAM map. In Figure 3, (a) and (b) show features detected at the beginning of a sequence where a camera makes a small loop around an office scene. In (c) and (d) we see these features again at the end of the loop, just before and just after loop-closing. Note the large search ellipses for the group of long-since observed 'old' features in (c), which are seen alongside a group of recently-initialised 'young' features in the left of the frame.

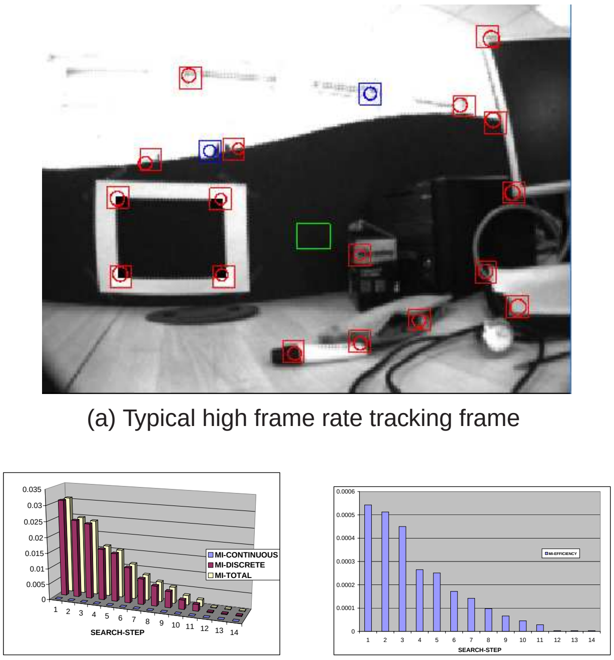
In the search prior for the frame shown in (c), there is high correlation within each of the two groups (because they have mostly co-occured in the frames when they were measured), but very weak correlation between the two groups for the same reason. Information efficiency directs active matching to measure the young features first, but measuring the features of the young group has a negligible effect on the search regions of the other group (red ellipses for features $F0, ..., F5$ are only marginally smaller than the black ones). Measuring then one of the old landmarks involves searching a large image region and we therefore encounter more than one template match candidates. This is where the multiple hypothesis power of active matching comes in to resolve this ambiguity by cross-checking for consensus in the rest of the features of the old group. This happens naturally through active matching and the search regions of the unmeasured features reduce in size as more matches are found. As a result, a globally consisted data association scenario is found and loop closure is propagated correctly into the map. The number of image processing operations needed to establish global consensus on this frame is 37% of the number needed to search all of the large ellipses and resolve via JCBB.

For this loop closure example, note that the measurement of old features was deliberately delayed until there were at least 4 such features present in the image. This is necessary at this stage, because in the case where there were only one loop-closing feature present it would be virtually uncorrelated with the rest of the features, and any match encountered would be jointly compatible with the rest (meaning that neither active matching nor JCBB would be able to resolve whether the single loop-closing feature was correctly matched). In the future, we hope to attack this delay in loop closing via an extended application of active matching. The algorithm should report the ambiguity of the search for consensus in such a situation by terminating with multiple highly-weighted Gaussian hypotheses. It should be possible to propagate these multiple hypotheses through time until they can be fully resolved when further 'old' features come back into the field of view.

C. Use in High Frame-Rate Tracking

Another experiment investigated the operation of active matching at the other end of the range of possible input priors. Here we used a high frame rate camera to grab images at and run MonoSLAM 200Hz as in [7]. The high frequency of the incoming frames here means that the predictions of the feature locations can be very accurate (less than 100 pixels squared in image area typically). Tracking at high frame rates is an interesting prospect because it can proceed as frequent, small corrections to the predictions of an increasingly useful motion model. The gains in information to be made per image are smaller, but when the image measurement process is active in nature it should also be possible to make it computationally cheap.

With the small search regions at 200Hz, there is little ambiguity involved within the matching process for each frame. Also, the small search regions are not as highly correlated relative to their sizes as at lower frame-rates: each measurement of a feature does not reduce the size of the already small other feature search ellipses much. For this reason there is only a small reduction in the total number of image



(a) Typical high frame rate tracking frame

(b)MI-scores (c)MI-Efficiency scores

Fig. 4. Active matching algorithm applied to feature search in high frame rate 3D tracking. (a) A typical frame of the sequence demonstrating the high accuracy of features' location predictions in the image. (b) and (c) Evolution of the maximal MI and MI efficiency scores through the search-steps of active matching within a typical frame. The MI-continuous scores are dominating the behavior of the MI-scores while the MI-discrete part has a negligible effect, which is a proof of mainly unambiguous scenario. Therefore, both total MI and MI efficiency values tail off smoothly.

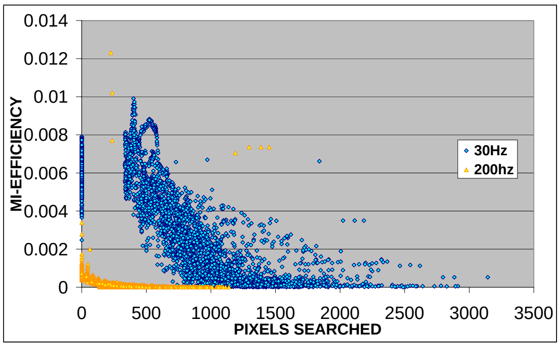


Fig. 5. The decay of MI-efficiency scores with the number of pixels searched for 30Hz (blue rhombs) and 200Hz (yellow triangles) tracking. The mutual information scores for high frame rate tracking at the beginning of matching within a frame is much lower and approaches negligible values much earlier (in terms of pixels searched) than in 30Hz tracking. This is due to the fact that tracking at such a high frame rate the uncertainty in the system is generally small, therefore localizing the first feature (less than 100 pixels searched) makes the uncertainty shrink to almost measurement error. On the other hand, tracking at 30Hz, the search regions of features are bigger and we need to localize 3-4 features before reducing the uncertainty in the system to a negligible value. The MI-efficiency score spikes are due to the ambiguities arising during matching to send search in a different direction. Such spikes are very rare in 200Hz tracking as generally there is not much ambiguity involved there.

processing operations when active matching is compared with independent searches of all feature ellipses and resolution via JCBB.

However, the active approach can make gains in a different way, by intelligently determining when in fact it is possible to *terminate* the matching process early and not measure all of the feature candidates available. A criterion can be set in terms of predicted information gain to determine when it is simply not worth measuring the other features since their locations can already be accurately predicted. We might imagine this criterion ultimately telling us that at super high frame-rates such as 1000Hz+ it is not worth measuring any features at all on some of the frames, effectively determining an optimal frame-rate for tracking motion of a certain level of dynamics.

Figure 4 demonstrates the behaviour of the mutual information and information efficiency scores per feature searched during 200Hz active matching, and Figure 5 is an experimental comparison of the information efficiency per pixel searched at 30Hz and 200Hz. It can be seen that the search process can be rigorously cut off at at much earlier point at the higher frame-rate to achieve the same level of tracking accuracy.

V. Conclusions

We have shown that a mixture of Gaussians formulation allows global consensus feature matching to proceed in a fully sequential, Bayesian algorithm which we call active matching. Information theory plays a key role in guiding highly efficient image search and we can achieve large factors in the reduction of image processing operations.

We plan to experiment with this algorithm in a range of different scenarios to gauge the effectiveness of active search at different frame-rates, resolutions, feature densities and tracking dynamics. While our initial instinct was that the algorithm would be most powerful in matching problems with strong priors such as high frame-rate tracking due to the advantage it can take of good predictions, our experiments with lower frame-rates indicate its potential also in other problems such as recognition. There priors on absolute feature locations will be weak but priors on relative locations may still be strong.

Acknowledgements

This research was supported by EPSRC grant GR/T24685/01. We are grateful to Ian Reid, José María Montiel, José Neira, Javier Civera and Paul Newman for very useful discussions, and to Peter Gemeiner for supplying the high frame-rate image sequences.

References

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

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

[3] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

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

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

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] P. Gemeiner, A. J. Davison, and M. Vincze. Improving localization robustness in monocular SLAM using a high-speed camera. In *Proceedings of Robotics: Science and Systems (RSS)*, 2008.

[8] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA: MIT Press, 1990.

[9] J. Manyika. *An Information-Theoretic Approach to Data Fusion and Sensor Management*. PhD thesis, University of Oxford, 1993.

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

[11] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalisation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2007.