Active Matching

Margarita Chli and Andrew J. Davison

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

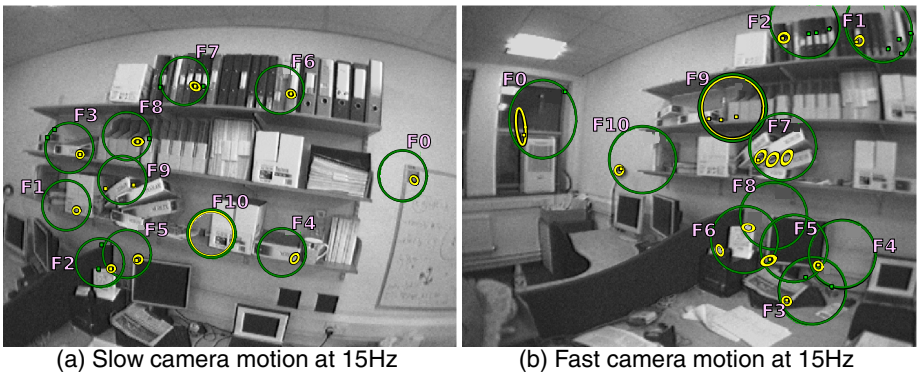
Abstract. In the matching tasks which form an integral part of all types of tracking and geometrical vision, there are invariably priors available on the absolute and/or relative image locations of features of interest. Usually, these priors are used post-hoc in the process of resolving feature matches and obtaining final scene estimates, via 'first get candidate matches, then resolve' consensus algorithms such as RANSAC. 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 and large search regions.

1 Introduction

It is well known that the key to obtaining correct feature associations in potentially ambiguous image matching 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.

Outliers are match candidates which lie outside of bounds determined by global consensus constraints: these are priors on the true absolute and relative locations of features if expressed in a proper probabilistic manner. 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. The approach in the active matching paradigm of this paper 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

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 72–85, 2008.



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

Fig. 1. Active matching dramatically reduces image processing operations while still achieving global matching consensus. Here, in a search for 11 point features in 3D camera tracking we contrast green regions for standard feature search with the much smaller yellow ellipses searched by our Active Matching method. In these frames, joint compatibility needed to search a factor of 4.8 more image area than Active Matching in (a) and a factor or 8.4 in (b). Moreover, JCBB encounters all the matches shown (blobs), whereas Active Matching only finds the yellow blobs.

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.

While matching is often formulated as a search for correspondence between one image and another (for example in the literature on 3D multi-view constraints with concepts such as the multi-view tensors), stronger constraints are available when we consider matching an image to a *state* — an estimate of world properties perhaps accumulated over many images. Uncertainty in a state is represented with a probability distribution. Matching constraints are obtained by projecting the 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 [11] which matches features via a deterministic interpretation tree [7] and has been applied to geometric image matching in [1]. 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 goes much further than

previously published 'guided matching' algorithms such as [12] in guiding not just a search for consensus but the image processing to determine candidate matches themselves.

Davison [3] presented a theoretical analysis of information gain in sequential image search. However, this 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.

2 Probabilistic Prediction and Feature by Feature Search

We consider making image measurements of an object or scene of which the current state of knowledge is modelled by a probability distribution over a finite vector of parameters $\mathbf{x}$ — representing the position of a moving object or camera, for instance. In an image, we are able to observe *features*: measurable projections of the scene state. A measurement of feature i yields the vector of parameters $\mathbf{z}_i$ — for example the 2D image coordinates of a keypoint. A likelihood function $p(\mathbf{z}_i|\mathbf{x})$ models the measurement process.

When a new image arrives, we can project the current probability distribution over state parameters $\mathbf{x}$ into feature space to *predict* the image locations of all the features which are measurement candidates. Defining stacked vector $\mathbf{z}_T = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \dots \end{pmatrix}^\top$ containing all candidate feature measurements, the density:

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

is a probabilistic prediction not just of the most likely image position of each feature, but a joint distribution over the expected locations of all of them. Given just individually marginalised parts $p(\mathbf{z}_i)$ of this prediction, the image search for each feature can sensibly be limited to high-probability regions, which will practically often be small in situations such as tracking. In Isard and Blake's Condensation [8], for example, feature searches take place in fixed-size windows around pre-determined measurement sites centred at a projection into measurement space of each of the particles representing the state probability distribution.

However, the extra information available that has usually been overlooked in feature search but which we exploit in this paper is that the predictions of the values of all the candidate measurements which make up joint vector $\mathbf{z}_T$ are often highly correlated, since they all depend on common parts of the scene state $\mathbf{x}$. In a nutshell, the correlation between candidate measurements means that making a measurement of one feature tells us a lot about where to look for another feature, suggesting a step by step guided search rather than blanket examination of all feature regions.

2.1 Guiding Search Using Information Theory

At each step in the search, the next feature and search region must be selected. Such candidate measurements vary in two significant ways: the amount of information which they are expected to offer, and the amount of image processing likely to be required to extract a match; both of these quantities can be computed directly from the current search prior. There are ad-hoc ways to score the value of a measurement such as search ellipse size, used for simple active search for instance in [5]. However, Davison [3], building on early work by others such as Manyika [10], explained clearly that the Mutual Information (MI) between a candidate and the scene state is the essential probabilistic measure of measurement value.

Following the notation of Mackay [9], the (MI) of continuous multivariate PDFs $p(\mathbf{x})$ and $p(\mathbf{z}_i)$ is:

$$I(\mathbf{x};\mathbf{z}_i) = E\left[\log_2 \frac{p(\mathbf{x}|\mathbf{z}_i)}{p(\mathbf{x})}\right] \tag{2}$$

$$= \int_{\mathbf{x},\mathbf{z}_i} p(\mathbf{x},\mathbf{z}_i)\log_2 \frac{p(\mathbf{x}|\mathbf{z}_i)}{p(\mathbf{x})} d\mathbf{x}d\mathbf{z}_i \ . \tag{3}$$

Mutual information is *expected information gain*: $I(\mathbf{x};\mathbf{z}_i)$ is how many **bits** of information we expect to learn about the uncertain vector $\mathbf{x}$ by determining the exact value of $\mathbf{z}_i$. In active matching, the MI scores of the various candidate measurements $\mathbf{z}_i$ can be fairly compared to determine which has most utility in reducing uncertainty in the state $\mathbf{x}$, even if the measurements are of different types (e.g. point feature vs. edge feature). Further, dividing MI by the computational cost required to extract a measurement leads to an 'information efficiency' score [3] representing the bits to be gained per unit of computation.

We also see here that when evaluating candidate measurements, a useful alternative to calculating the mutual information $I(\mathbf{x};\mathbf{z}_i)$ between a candidate measurement and the state is to use the MI $I(\mathbf{z}_{T\neq i};\mathbf{z}_i)$ between the candidate and *all the other candidate measurements*. This is a measure of how much information the candidate would provide about the other candidates, capturing the core aim of an active search strategy to decide on measurement order. This formulation has the very satisfying property that active search can proceed purely in measurement space, and is appealing in problems where it is not desirable to make manipulations of the full state distribution during active search.

2.2 Active Search Using a Single Gaussian Model

To attack the coupled search problem, Davison [3] made the simplifying assumption that the PDFs describing knowledge of $\mathbf{x}$ and $\mathbf{z}_T$ can be approximated always by single multi-variate Gaussian distributions. The measurement process is modelled by $\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}) + \mathbf{n}_m$, where $\mathbf{h}_i(\mathbf{x})$ describes the functional relationship between the expected measurement and the object state as far as understood via the models used of the object and sensor, and $\mathbf{n}_m$ is a Gaussian-distributed vector representing unmodelled effects (noise) with covariance $\mathtt{R}_i$ which is independent for each measurement. The vector $\mathbf{x}_m$ which stacks the object state and candidate measurements (in measurement space) can be calculated along with its full covariance:

$$\hat{\mathbf{x}}_m = \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}} \\ \mathbf{h}_1(\hat{\mathbf{x}}) \\ \mathbf{h}_2(\hat{\mathbf{x}}) \\ \vdots \end{pmatrix}, \; \mathtt{P}_{\mathbf{x}_m} = \begin{bmatrix} \mathtt{P}_x & \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top & \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top & \cdots \\ \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top + \mathtt{R}_1 & \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top & \cdots \\ \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}}^\top & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathtt{P}_x \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}}^\top + \mathtt{R}_2 & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix} \quad (4)$$

The lower-right portion of $\mathtt{P}_{\mathbf{x}_m}$ representing the covariance of $\mathbf{z}_T = \begin{pmatrix} \mathbf{z}_1 \; \mathbf{z}_2 \; \ldots \end{pmatrix}^\top$ is known as the *innovation covariance matrix* $\mathtt{S}$ in Kalman filter tracking. The correlations between different candidate measurements mean that generally $\mathtt{S}$ will not be block-diagonal but contain off-diagonal correlations between the predicted measurements of different features.

With this single Gaussian formulation, the mutual information in bits between any two partitions α and β of $\mathbf{x}_m$ can be calculated according to this formula:

$$I(\alpha;\beta) = \frac{1}{2}\log_2 \frac{|\mathtt{P}_{\alpha\alpha}|}{|\mathtt{P}_{\alpha\alpha} - \mathtt{P}_{\alpha\beta}\mathtt{P}_{\beta\beta}^{-1}\mathtt{P}_{\beta\alpha}|}\,, \quad (5)$$

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:

$$I(\alpha;\beta) = H(\alpha) - H(\alpha|\beta) = H(\alpha) + H(\beta) - H(\alpha,\beta) \quad (6)$$

$$= \frac{1}{2}\log_2 \frac{|\mathtt{P}_{\alpha\alpha}||\mathtt{P}_{\beta\beta}|}{|\mathtt{P}_{\mathbf{x}_m}|}\,. \quad (7)$$

2.3 Multiple Hypothesis Active Search

The weakness of the single Gaussian approach of the previous section is that, as ever, a Gaussian is uni-modal and can only represent a PDF with one peak. 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. This is the motivation for the mixture of Gaussians formulation used in our active matching algorithm. We wish to retain the feature-by-feature quality of active search. The 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.

3 Active Matching Algorithm

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.

3.1 Search State Mixture of Gaussians Model

A single multi-variate Gaussian probability distribution over vector $\mathbf{x}_m$ defined in Equation 4 is parameterised by a 'mean vector' $\hat{\mathbf{x}}_m$ and covariance matrix $\mathtt{P}_{\mathbf{x}_m}$, and we use the shorthand $\mathbf{G}(\hat{\mathbf{x}}_m, \mathtt{P}_{\mathbf{x}_m})$ to represent the explicit normalised PDF:

$$p(\mathbf{x}_m) = \mathbf{G}(\hat{\mathbf{x}}_m, \mathtt{P}_{\mathbf{x}_m}) \tag{8}$$

$$= (2\pi)^{-\frac{D}{2}} |\mathtt{P}_{\mathbf{x}_m}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_m - \hat{\mathbf{x}}_m)^\top \mathtt{P}_{\mathbf{x}_m}^{-1}(\mathbf{x}_m - \hat{\mathbf{x}}_m)} . \tag{9}$$

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:

$$p(\mathbf{x}) = \sum_{i=1}^{K} p(\mathbf{x}_i) = \sum_{i=1}^{K} \lambda_i \mathbf{G}_i \ , \tag{10}$$

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. We normally assume that the input prior at the start of the search process is well-represented by a single Gaussian and therefore $\lambda_1 = 1, \lambda_{i \neq 1} = 0$. 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 $\sum_{i=1}^{K} \lambda_i = 1$.

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 3.4.

3.2 The Algorithm

The active matching algorithm (see Figure 2) is initialized with a joint Gaussian prior over the features' locations in measurement space (e.g. prediction after application of motion model). At each step we select a $\{$Gaussian, Feature$\}$ pair for measurement based on the expected information gain (see Section 4) and make an exhaustive search for feature matches within this region, finding zero or more matches above a threshold. For every template match yielded by the search a new Gaussian is spawned with mean and covariance conditioned on the hypothesis of that match being a true positive, and we also consider the 'null' possibility that none of the matches is a true positive. After a search the MoG distribution is updated to represent the outcome, as detailed in the rest of this section. Very weak Gaussians are pruned from the mixture after each search step. The algorithm continues until all features have been measured, or an alternative stopping criterion can be defined based on expected information gain falling below a desired value indicating that nothing more of relevance is to be obtained from the image.

3.3 Likelihood Function

One step of active matching 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:

ACTIVEMATCHING($\mathbf{G}_{in}$)
Mixture = $[[1, \mathbf{G}_{in}]]$
$\{\mathbf{F}_{sel}, \mathbf{G}_{sel}\}$ = get_max_gain_pair(Mixture)
while is_unmeasured($\mathbf{F}_{sel}, \mathbf{G}_{sel}$)
Matches = measure($\mathbf{F}_{sel}, \mathbf{G}_{sel}$)
UPDATEMIXTURE(Mixture,Matches)
prune_weak_gaussians(Mixture)
$\{\mathbf{F}_{sel}, \mathbf{G}_{sel}\}$ = get_max_gain_pair(Mixture)
end while
$\mathbf{G}_{best}$ = get_most_probable_gaussian(Mixture)
return $\mathbf{G}_{best}$

UPDATEMIXTURE(Mixture$_{1..K}$,Matches$_{1..M}$)
$[\lambda_i, \mathbf{G}_i]$ = get_measured_gaussian(Mixture)
for $m = 1 : M$
$[\lambda_m, \mathbf{G}_m]$ = fuse_match($\mathbf{G}_i$,λ_i,Matches[m])
Mixture = [Mixture,$[\lambda_m, \mathbf{G}_m]$]
end for
for $k = 1 : K$
$\lambda_{k,new}$ = update_weight(λ_k,Matches)
Mixture[k] = $[\lambda_{k,new}, \mathbf{G}_k]$
end for
normalize_weights(Mixture)
return

Fig. 2. Active matching algorithm and UPDATEMIXTURE sub-procedure

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:

$$p(\mathbf{z}_c|\mathbf{x}) = \mu_{\text{in}}\mathbf{T}_{\text{in}} + \mu_{\text{out}}\mathbf{T}_{\text{out}} + \sum_{\mathbf{m=1}}^{M} \mu_{\text{match}}\mathbf{H_m} \, . \tag{11}$$

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

$$\mu_{\text{in}} = P_{\mathbf{fp}}^{M} P_{\mathbf{fn}} P_{\mathbf{tn}}^{N-(M+1)} \tag{12}$$

$$\mu_{\text{out}} = P_{\mathbf{fp}}^{M} P_{\mathbf{tn}}^{N-M} \tag{13}$$

$$\mu_{\text{match}} = P_{\mathbf{tp}} P_{\mathbf{fp}}^{M-1} P_{\mathbf{tn}}^{N-M} \, , \tag{14}$$

where $P_{\mathbf{tp}}$, $P_{\mathbf{fp}}$ $P_{\mathbf{tn}}$, $P_{\mathbf{fn}}$ are per-pixel 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.

3.4 Posterior: Updating After a Measurement

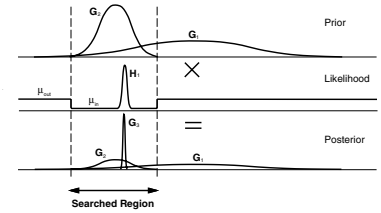
The standard application of Bayes' Rule to obtain the posterior distribution for $\mathbf{x}$ given the new measurement is:

$$p(\mathbf{x}|\mathbf{z}_c) = \frac{p(\mathbf{z}_c|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}_c)} \, . \tag{15}$$

Substituting MOG models from Equations 10 and 11:

$$p(\mathbf{x}|\mathbf{z}_c) = \frac{\left(\mu_{\text{in}}\mathbf{T}_{\text{in}} + \mu_{\text{out}}\mathbf{T}_{\text{out}} + \sum_{\mathbf{m=1}}^{M} \mu_{\text{match}}\mathbf{H}_{\mathbf{m}}\right)\left(\sum_{i=1}^{K} \lambda_i \mathbf{G}_i\right)}{p(\mathbf{z}_c)} . \tag{16}$$

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 16, 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.

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 far 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 [11].

4 Measurement Selection

4.1 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 K is the

number of Gaussians and 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:

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{G}_1$	$*$			

$\Rightarrow$

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{G}_1$	$*$			
$\mathbf{G}_2$	$*$		$*$	

$\Rightarrow$

	$\mathbf{F}_1$	$\mathbf{F}_2$	$\mathbf{F}_3$	$\mathbf{F}_4$
$\mathbf{G}_1$	$*$			
$\mathbf{G}_2$	$*$		$*$	
$\mathbf{G}_3$	$*$		$*$	

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 [3] which is this mutual information divided by the area of the search region, assumed proportional to search cost.

4.2 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 (the desire to *reduce ambiguity*), which (b) $I_{\text{continuous}}$ gives a measure of the reduction in covariance of the most likely Gaussian which becomes more peaked after a match (the desire to *increase precision*). Due to the intuitive absolute nature of mutual information, these terms are additive:

$$I = I_{\text{discrete}} + I_{\text{continuous}} \tag{17}$$

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.

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 it is predicted to provide by:

$$I(\mathbf{x};\mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}). \tag{18}$$

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

$$I_{\text{discrete}} = H(\mathbf{x}) - P(\mathbf{z} = \text{null}) \quad H(\mathbf{x}|\mathbf{z} = \text{null}) \tag{19}$$

$$- P(\mathbf{z} = \text{match}) H(\mathbf{x}|\mathbf{z} = \text{match}) \, . \tag{20}$$

where

$$H(\mathbf{x}) = \sum_{i=1}^{K} \lambda_i \log_2 \frac{1}{\lambda_i}, \; H(\mathbf{x}|\mathbf{z} = \text{null}) = \sum_{i=1}^{K} \lambda'_i \log_2 \frac{1}{\lambda'_i}, \; H(\mathbf{x}|\mathbf{z} = \text{match}) = \sum_{i=1}^{K+1} \lambda''_i \log_2 \frac{1}{\lambda''_i}. \tag{21}$$

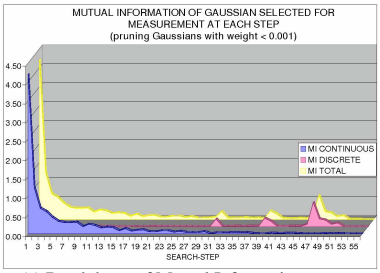
The predicted weights after a null or a match search are calculated as in Equation 16 with the only difference that the likelihood of a match-search is summed over all positions in the search-region that can possibly yield a match.

Mutual Information: Continuous Component. Continuous MI is computed using Equation 7:

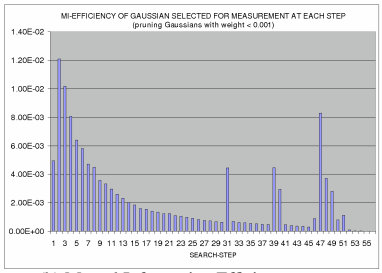
$$I_{\text{continuous}} = \frac{1}{2} P(\mathbf{z} = \text{match}) \lambda''_m \log_2 \frac{|\mathtt{P}_{\alpha\alpha}||\mathtt{P}_{\beta\beta}|}{|\mathtt{P}_{\mathbf{x}_m}|} \tag{22}$$

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 [3] but is now scaled and combined with discrete component arising due to the expected change in the λ_i distribution. As explained in Section 2, we can replace the product $|\mathtt{P}_{\alpha\alpha}||\mathtt{P}_{\beta\beta}|$ with $|\mathtt{P}_{\mathbf{z}_{T\neq i}}||\mathtt{P}_{\mathbf{z}_{T=i}}|$ to calculate a continuous MI score in measurement space.

Figure 3(a, b) shows the MI and MI efficiency scores of the selected measurement at each step of the matching process when Active Matching is applied to a frame from MonoSLAM (see Section 5) with around 50 candidate features. These plots demonstrate the expected tailing off of measurement utility and the diminishing returns of continued search.



(a) Breakdown of Mutual Information scores



(b) Mutual Information Efficiency scores

Fig. 3. The evolution of MI and MI-efficiency scores of the selected measurement through the search-steps of Active Matching within a frame, tracking on average 50 features. Both values tail off generally with spikes as null searches or ambiguities arise and send search in a different direction. In (a) the total Mutual Information is shown broken down into its discrete and continuous components. It is the continuous component which displays a smooth decay with search step number, while the discrete component spikes up at ambiguities.

5 Results

We present results on the application of the algorithm to feature matching within the publically available MonoSLAM system [4] for real-time probabilistic structure and motion estimation. This system, which is well known for its computational efficiency thanks to 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 at 15 or 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 [1,13] 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 [2]. An example of how ambiguity is handled and resolved by active matching within a typical MonoSLAM frame is shown in Figure 4.

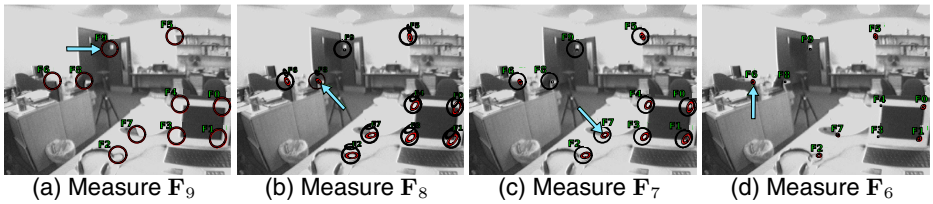

(a) Measure $\mathbf{F}_9$ (b) Measure $\mathbf{F}_8$ (c) Measure $\mathbf{F}_7$ (d) Measure $\mathbf{F}_6$

Fig. 4. 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.

5.1 Sequence Results

Two different hand-held camera motions were used to capture image sequences at 30Hz: one with a standard level of dynamics slightly faster than in the results of [4], and one with much faster, jerky motion (see the video submitted with this paper). 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. 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. This is 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.

This new real-time ability to tracking extremely rapid camera motion at a range of frame-rates significantly expands the potential applications of 3D camera tracking. Please see the submitted videos for full illustration of the operation of active matching on these sequences.

5.2 Computational Complexity

We have seen that active matching will always reduce the number of image processing operations required when compared to blanket matching schemes, but it requires extra computation in calculating *where to search* at each step of the matching process. The sequence results indicate that these extra computations are more than cancelled out by the gain in image processing speed, but it is appropriate to analyse of their computational complexity.

Each step of the active matching algorithm first requires MI efficiency scores to be generated and compared for up to the KF measurable combinations of feature and current live Gaussians.

Each MI evaluation requires computation of order $O(K)$ for the discrete component and $O(F^3)$ for the continuous component using formula Equation 22 (the deterimnants can be computed by LU decomposition or similar). The constants of proportionality are small here and these evaluations are cheap for low numbers of feature candidates. Although the the cost of evaluating continuous MI scales poorly with the number of feature candidates, in practice if the image feature density is high then it will be sensible to limit the number of candidates selected between at each step: for instance one candidate could be randomly chosen from each block of a regular grid overlaid on the image, on the assumption that candidates within a small region are highly correlated and choosing between them is unnecessary.

The number of steps required to achieve global matching of all features will be around $\bar{K}F$, where $\bar{K}$ is the average number of live Gaussians after pruning. However, in practical applications with large numbers of features we will be able to improve on this by terminating the matching process when the expected information gain from any remaining candidates drops below a threshold — again, when the feature density is very high, there will be many highly correlated feature candidates and the mutual information criterion will tell us that there is little point in measuring all of them.

6 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 useful discussions.

References

1. Clemente, L.A., Davison, A.J., Reid, I.D., Neira, J., Tardós, J.D.: Mapping large loops with a single hand-held camera. In: Proceedings of Robotics: Science and Systems (RSS) (2007)
2. Cummins, M., Newman, P.: Probabilistic appearance based navigation and loop closing. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2007)
3. Davison, A.J.: Active search for real-time vision. In: Proceedings of the International Conference on Computer Vision (ICCV) (2005)
4. Davison, A.J., Molton, N.D., Reid, I.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29(6), 1052–1067 (2007)
5. Davison, A.J., Murray, D.W.: Mobile robot localisation using active vision. In: Proceedings of the European Conference on Computer Vision (ECCV) (1998)
6. Fischler, M.A., Bolles, R.C.: 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. Grimson, W.E.L.: *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge (1990)
8. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: Proceedings of the 4th European Conference on Computer Vision (ECCV), Cambridge, pp. 343–356 (1996)
9. Mackay, D.: *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge (2003)
10. Manyika, J.: An Information-Theoretic Approach to Data Fusion and Sensor Management. PhD thesis, University of Oxford (1993)
11. Neira, J., Tardós, J.D.: Data association in stochastic mapping using the joint compatibility test. IEEE Trans. Robotics and Automation 17(6), 890–897 (2001)
12. Tordoff, B., Murray, D.: Guided-MLESAC: Faster image transform estimation by using matching priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27(10), 1523–1535 (2005)
13. Williams, B., Klein, G., Reid, I.: Real-time SLAM relocalisation. In: Proceedings of the International Conference on Computer Vision (ICCV) (2007)