

Active matching for visual tracking

Margarita Chli*, Andrew J. Davison

Department of Computing, Imperial College London, London SW7 2AZ, UK

ARTICLE INFO

Article history:

Available online 24 July 2009

Keywords:

Image matching
Visual tracking
SLAM

ABSTRACT

In the feature matching tasks which form an integral part of visual tracking or SLAM (Simultaneous Localisation And Mapping), 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 far 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 denoted Active Matching, 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 with a range of settings, and give a detailed analysis of performance which leads to performance-enhancing approximations to the full algorithm.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

It is well known that the key to obtaining correct feature associations in potentially ambiguous matching (data association) tasks using computer vision or other sensors 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 of a certain type of salient features in a pair of images and pairing up features which have similar appearance descriptors. 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 random sampling and voting algorithm RANSAC [1] 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. 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. Both individual feature motion assumptions (such as that the image displacement of a feature between consecutive video frames will be bounded) *and* global consensus constraints can be expressed as priors on the true absolute and relative locations of features within a rigorous Bayesian framework.

In Active Matching, instead of searching for all features and then resolving, feature searches occur one by one within tightly targeted regions. The results of each search 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*. The result is that the number of image processing operations required to achieve global matching is reduced by a large factor.

We show that information theory is able to intelligently guide the step by step search process and answer the question “where to look next?”. The expected information content of each candidate measurement is computed and compared, and can also be traded off against the expected computational cost of the image processing required. The absolute bit units of information scores mean that heterogeneous feature types can be rigorously and intuitively combined within the same matching process. Information theory can also indicate when matching should be terminated at a point of diminishing returns.

* Corresponding author.

E-mail addresses: mchli@doc.ic.ac.uk (M. Chli), ajd@doc.ic.ac.uk (A.J. Davison).

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 random sampling and preset thresholds of RANSAC are unsatisfying. In more recent variants of the algorithm it has been realised that an unnecessarily large number of association hypotheses gets tested, therefore speedups have been proposed either by a two-step randomised selection of hypotheses [2] or taking some motion priors into account [3,4]. However, the true value of the probabilistic priors available has not yet fully been appreciated and exploited in these methods which rely heavily on randomness and arbitrary thresholds. This has been improved by probabilistic methods such as the Joint Compatibility Branch and Bound (JCBB) algorithm [5] which matches features via a deterministic interpretation tree [6] and has been applied to geometric image matching in [7]. JCBB takes account of a joint Gaussian prior on feature positions and calculates the joint probability that any particular hypothesised 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 [4] in guiding not just a search for consensus but the image processing to determine candidate matches themselves.

Davison [8] 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.

In this paper we present the Active Matching algorithm (first introduced in [9,10]) in full detail. We explain more clearly the motivation for the mixture representation with a new histogram-based analysis of the underlying probability distributions. We also include a comprehensive new set of experiments which examines the performance of the algorithm in monocular structure and motion tracking as parameters including frame-rate and feature density are varied. These experiments indicate the route to effective approximations which further increase the efficiency of the algorithm.

2. Probabilistic prediction and feature by feature search

In our general matching formulation, 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} . These parameters may represent the position of a moving object or camera, for instance. The probability distribution $p(\mathbf{x})$ which describes our uncertain knowledge of the parameters at the moment an image arrives will be determined by

general prior knowledge and what has happened previously to the system. For instance, in the common case of sequential tracking of motion through an image sequence, $p(\mathbf{x})$ at each time step will be the result of projecting the distribution determined at the previous frame forward through a motion model.

In an image, we are able to observe *features*: measurable projections of the state. A measurement of feature i yields the vector of parameters \mathbf{z}_i . For example, \mathbf{z}_i might be the 2D image coordinates of a keypoint of known appearance, the position of an edge or a higher-dimensional parameterisation of a more complex image entity. In each case, 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 $\mathbf{z}_T = (\mathbf{z}_1 \ \mathbf{z}_2 \ \dots)^T$ containing all candidate feature measurements and stacked likelihood function $p(\mathbf{z}_T|\mathbf{x})$, the density:

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

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. This joint distribution, if formulated correctly, takes full account of both individual feature motion assumptions and global inter-feature constraints.

Our goal is to use $p(\mathbf{z}_T)$ to guide intelligent active search and matching. The first possibility one might consider is to marginalise elements $p(\mathbf{z}_i)$ to give individual predictions of the image location of each feature under consideration. Image search for each feature can then sensibly be limited to high-probability regions. This procedure is relatively common in visual tracking, where strong motion models mean that these search regions are often small and efficiently searched. In Isard and Blake’s Condensation [11], 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. Several Kalman Filter-based trackers such as [12] implement the same scheme by using gates at a certain number of standard deviations to restrict the search.

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 [13]. However, Davison [8], building on early work by others such as Manyika [14], 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 [15], the (MI) of continuous multivariate PDFs $p(\mathbf{x})$ and $p(\mathbf{z}_i)$ is:

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

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

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 [8] 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 [8] made the simplifying assumption that the PDFs describing the 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 \mathbf{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:

$$\hat{\mathbf{x}}_m = \begin{pmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{z}}_1 \\ \hat{\mathbf{z}}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{x}} \\ \mathbf{h}_1(\hat{\mathbf{x}}) \\ \mathbf{h}_2(\hat{\mathbf{x}}) \\ \vdots \end{pmatrix}, \quad (4)$$

$$\mathbf{P}_{\mathbf{x}_m} = \begin{bmatrix} \mathbf{P}_x & & & & \\ \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} \mathbf{P}_x & \mathbf{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} + \mathbf{R}_1 & & & \\ \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathbf{P}_x & \frac{\partial \mathbf{h}_2}{\partial \mathbf{x}} \mathbf{P}_x \frac{\partial \mathbf{h}_1}{\partial \mathbf{x}} + \mathbf{R}_2 & & & \\ \vdots & \vdots & \vdots & \vdots & \end{bmatrix}$$

The lower-right portion of $\mathbf{P}_{\mathbf{x}_m}$ representing the covariance of $\mathbf{z}_T = (\mathbf{z}_1 \ \mathbf{z}_2 \ \dots)^\top$ is known as the *innovation covariance matrix* \mathbf{S} in Kalman filter tracking. The correlations between different candidate measurements mean that generally \mathbf{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:

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

where $\mathbf{P}_{\alpha\alpha}$, $\mathbf{P}_{\alpha\beta}$, $\mathbf{P}_{\beta\beta}$ and $\mathbf{P}_{\beta\alpha}$ are sub-blocks of $\mathbf{P}_{\mathbf{x}_m}$. 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:

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

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

2.3. Multiple hypothesis active search using full histograms

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.

To investigate the theoretical performance of active search in such ambiguous cases, we developed a simulation of 1D Bayesian active search for a single feature which uses a simple but exhaustive histogram representation of probability (see Fig. 1). The goal is to locate a feature in a one-dimensional search region by making pixel-by-pixel attempts at template matching. Each pixel is represented by a discrete histogram bin storing the current probability that the true feature is in that location. The true feature must lie in exactly one true position, so at all times the discrete histogram is normalised to total probability one. At the start of search, we initialise a Gaussian prior across the region.

Active search proceeds by selecting pixel location i as a candidate, attempting a template match to achieve either a match M_i or failure F_i , and updating the whole histogram via Bayes rule. The update uses the following likelihood expression:

$$P(M_i|B_k) = C_{FP} + C_{TP} e^{-\frac{1}{2} \frac{(i-k)^2}{\sigma^2}} \quad (8)$$

$$P(F_i|B_k) = 1 - P(M_i|B_k) \quad (9)$$

for the probabilities of making a template match M_i or failed match F_i at position i given B_k , that the feature is truly at position k . Here C_{FP} is a constant representing the per-pixel false-positive probability of finding a template match to clutter, and C_{TP} is a constant proportional to the true-positive probability of matching the feature in its true position. This likelihood function says that if the feature is at k there is a raised, Gaussian-profile probability of making a match at nearby locations, the parameter σ (with a low value of one pixel or less) specifying the standard deviation of the feature’s ‘measurement uncertainty’.

The last figure here is the motivation for the mixture of Gaussians formulation used in the rest of the paper. The single Gaussian method of Section 2.2 cannot represent the clear multiple hypotheses present here. This histogram representation really gets to the truth of active search, but is impractical in reality because of the computational cost of maintaining a histogram – rising exponentially with the number of dimensions of the total measurement vector. Practical real-time searches happen not by one-by-one pixel checks followed by probabilistic updates, but by examining a whole region at once and obtaining zero, one or more candidate matches. Fig. 1(d) shows that a mixture of Gaussians represents the posterior in this case well.

3. Active matching

Ideally, any features selected for measurement would be absolutely unique and always recognisable, meaning that they produce

a match only when present and at the true feature location. Since this is not the case in real image search problems, we can never fully trust the matching outcome of a feature search. Modelling the probabilistic ‘search state’ as a mixture of Gaussians, we wish to retain the feature-by-feature quality of active search [8]. Our new MoG representation allows dynamic, online updating of the multi-peaked PDF over feature locations which represents the multiple hypotheses arising 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.

3.1. 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 $\mathbf{P}_{\mathbf{x}_m}$. We use the shorthand $\mathbf{G}(\hat{\mathbf{x}}_m, \mathbf{P}_{\mathbf{x}_m})$ to represent the explicit normalised PDF:

$$p(\mathbf{x}_m) = \mathbf{G}(\hat{\mathbf{x}}_m, \mathbf{P}_{\mathbf{x}_m}) \quad (10)$$

$$= (2\pi)^{-\frac{D}{2}} |\mathbf{P}_{\mathbf{x}_m}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbf{x}_m - \hat{\mathbf{x}}_m)^\top \mathbf{P}_{\mathbf{x}_m}^{-1} (\mathbf{x}_m - \hat{\mathbf{x}}_m)}. \quad (11)$$

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 :

$$p(\mathbf{x}) = \sum_{i=1}^K p(\mathbf{x}_i) = \sum_{i=1}^K \lambda_i \mathbf{G}_i, \quad (12)$$

where we have now used the further notational shorthand $\mathbf{G}_i = \mathbf{G}(\hat{\mathbf{x}}_{m_i}, \mathbf{P}_{\mathbf{x}_{m_i}})$. 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 the next step of Active Matching. This prior together with the likelihood and posterior distributions which are shown in symboling 1D form in Section 3.4, are explained in the following sections.

3.2. The Algorithm

The MoG Active Matching process is initialised 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 {Feature, Gaussian} 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 4.

ACTIVEMATCHING(\mathbf{G}_0)

```

1 Mixture = [[1,  $\mathbf{G}_0$ ]] // consists of [weight, Gaussian] tuples
2 [ $f_c$ ,  $\mathbf{G}_c$ ] = get_max_gain_candidate(Mixture)
3 while (pair_not_yet_measured( $f_c$ ,  $\mathbf{G}_c$ ))
4   Matches = measure( $f_c$ ,  $\mathbf{G}_c$ )
5   UpdateMixture(Mixture,  $c$ ,  $f_c$ , Matches)
6   prune_insignificant_gaussians(Mixture)
7   [ $f_c$ ,  $\mathbf{G}_c$ ] = get_max_gain_candidate(Mixture)
8 end while
9  $\mathbf{G}_{best}$  = find_most_probable_gaussian(Mixture)
10 return  $\mathbf{G}_{best}$ 

```

For every template match yielded by the search of the selected {Feature, Gaussian} 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 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 the 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 [5].

UPDATEMIXTURE(Mixture, i , f , Matches)

Propagate the result of measuring feature f in \mathbf{G}_i in the Mixture, following the update rule of Eq. (18)

```

1 [ $\lambda_i$ ,  $\mathbf{G}_i$ ] = Mixture[ $i$ ]
2 for  $k = 1 : K$ 
3   // loop through all Gaussians to update them accordingly
4   if  $k = i$  then // this is the measured Gaussian
5     for  $m = 1 : M$  // for every match, spawn a new Gaussian
6        $\mathbf{G}_m = \text{spawn\_gaussian\_and\_fuse\_match}(\mathbf{G}_i, \text{Matches}[m])$ 
7        $\lambda_m = \lambda_i \times \mu_{\text{match}} \times \text{prior}(\text{Matches}[m], \mathbf{G}_i)$ 
8       Mixture = [Mixture, [ $\lambda_m$ ,  $\mathbf{G}_m$ ]]
9     end for
10     $\lambda_i = \lambda_i \times \mu_{\text{in}} \times (1 - \text{prior\_sum}(\text{Matches}, \mathbf{G}_i))$ 
11    Mixture[ $i$ ] = [ $\lambda_i$ ,  $\mathbf{G}_i$ ]
12  else
13    // Total probability of  $\mathbf{G}_k$  in the region covered by  $\mathbf{G}_i$ :
14    prob = prior_sum_under $\mathbf{G}_i(\mathbf{G}_k)$ 
15    sum = prior_sum(Matches,  $\mathbf{G}_k$ )
16     $\lambda_k = \lambda_k \times [\mu_{\text{match}} \times \text{sum} + \mu_{\text{in}}$ 
17       $\times (\text{prob} - \text{sum}) + \mu_{\text{out}} \times (1 - \text{prob})]$ 
18    Mixture[ $k$ ] = [ $\lambda_k$ ,  $\mathbf{G}_k$ ]
19  end if
20 end for
21 normalize_weights(Mixture)
22 return

```

Note: prior(Matches[m], \mathbf{G}_i) returns the prior probability of that match in \mathbf{G}_i (highest value at the centre of this gaussian). Similarly, prior_sum(Matches, \mathbf{G}) returns the sum of all such prior probabilities at the positions in Matches.

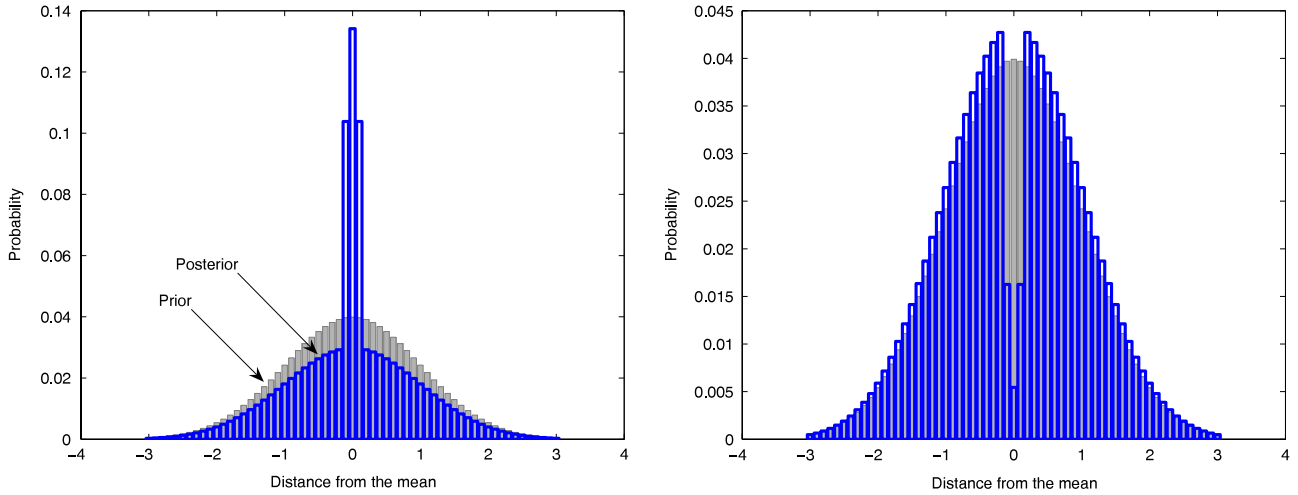
3.3. Likelihood function

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 f . Suppose that $\mathbf{z}_f = (\mathbf{z}_{f_1} \dots \mathbf{z}_{f_M})^\top$ is the outcome of this search for matches, meaning that template matching has been successful at M candidate pixel locations but failed everywhere else in the region. The likelihood $p(\mathbf{z}_f | \mathbf{x})$ of this result is modelled as a mixture consisting of:

$$p(\mathbf{z}_f | \mathbf{x}) = \mu_{\text{in}} \mathbf{T}_{\text{in}} + \mu_{\text{out}} \mathbf{T}_{\text{out}} + \sum_{m=1}^M \mu_{\text{match}} \mathbf{H}_m. \quad (13)$$

- M Gaussians \mathbf{H}_m , each representing the hypothesis of one candidate being the true **match** (considering all others as false

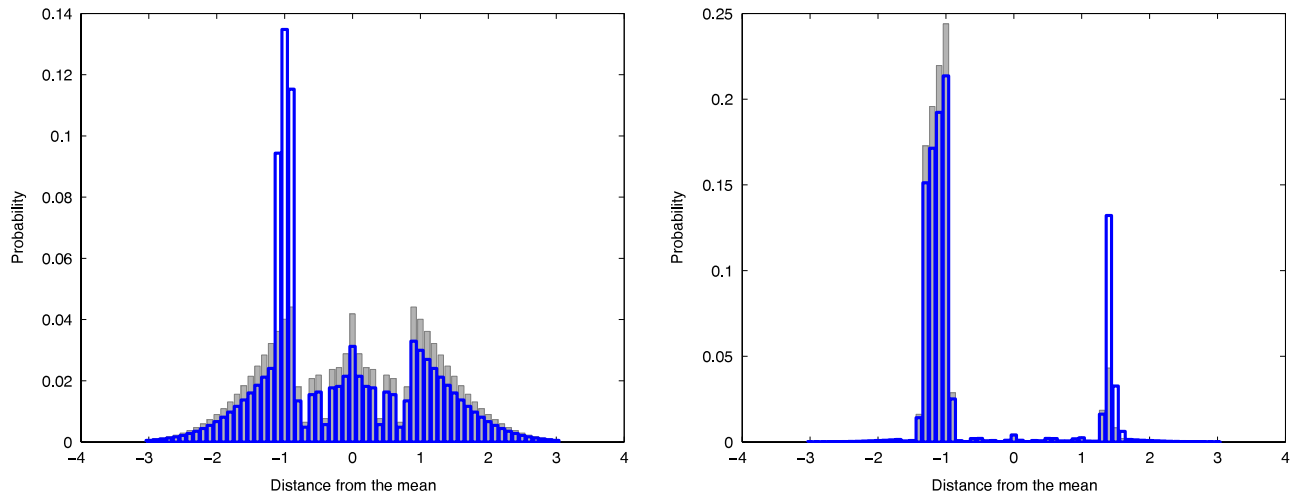
Test for a template match at central pixel:



(a) Success: match found

(b) Failure: no match present

Search state at later stages:



(c) 18 positions measured

(d) All positions measured

Fig. 1. One-dimensional pixel-by-pixel feature search. A full normalised histogram representing the probability that a feature is truly at each image location is refined sequentially from an initial Gaussian prior as each pixel location is tested for a template match. (a) and (b) show the outcome of either a successful or failed match at the pixel in the centre of the prior which is checked first: a success causes a spike in the distribution and a failure a trough. In (c), measurements at a number of central sites have led to an intermediate distribution, and (d) shows the final posterior distribution in a situation where all positions have been checked to reveal two significant candidate locations for the true feature, motivating our search state formulation as a mixture of Gaussians.

positives) – these Gaussians are functions of \mathbf{x} having the width of the measurement uncertainty R_i , and

- Two constant terms: \mathbf{T}_{in} representing the hypothesis that the true match lies **in** the searched region but has not been recognised, and \mathbf{T}_{out} supporting that the true feature is actually **out** of the region searched. Thus, both of these hypotheses consider all of the obtained matches as spurious false positives.

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

$$\mu_{in} = P_{fp}^M P_{fn} P_{tn}^{N-(M+1)} \quad (14)$$

$$\mu_{out} = P_{fp}^M P_{tn}^{N-M} \quad (15)$$

$$\mu_{match} = P_{tp} P_{fp}^{M-1} P_{tn}^{N-M}, \quad (16)$$

where $P_{tp}, P_{fp}, P_{tn}, P_{fn}$ are true-positive, false-positive, true-negative and false-negative probabilities respectively for this feature. \mathbf{T}_{in}

and \mathbf{T}_{out} are top-hat functions with value one inside and outside of the searched Gaussian 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} is the probability of obtaining M false positives, a false negative and $N - (M + 1)$ true negatives. μ_{out} is the probability of M false positives and $N - M$ true negatives. Finally, μ_{match} is the probability of a true positive occurring along with $M - 1$ 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:

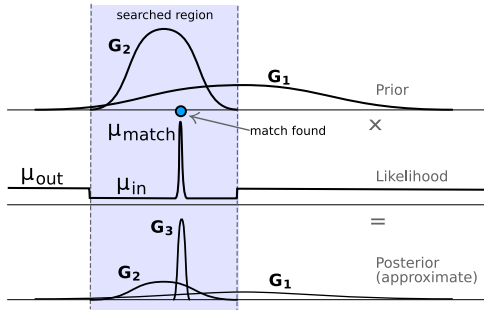
$$p(\mathbf{x}|\mathbf{z}_f) = \frac{p(\mathbf{z}_f|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}_f)}. \quad (17)$$

Substituting MoG models from Eqs. (12) and (13):

$$p(\mathbf{x}|\mathbf{z}_f) = \frac{\left(\mu_{\text{in}} \mathbf{T}_{\text{in}} + \mu_{\text{out}} \mathbf{T}_{\text{out}} + \sum_{m=1}^M \mu_{\text{match}} \mathbf{H}_m \right) \left(\prod_{i=1}^K \lambda_i \mathbf{G}_i \right)}{p(\mathbf{z}_f)}. \quad (18)$$

The denominator $p(\mathbf{z}_f)$ is a constant determined by normalising all new weights λ_i to add up to one). Below, is an illustration of the formation of a posterior when the search outcome consists of a single match ($M = 1$). This posterior will then become the prior for the next Active Matching step.

In the top line of Eq. (18), 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 information of making a null 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.

4. 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.

4.1. Search candidates

At each step of the MoG Active Matching process, we use the mixture to predict individual feature measurements, and there are KF possible actions, where F is the number of measurable features. We rule out any {Feature, Gaussian} 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 the selection:

$$\begin{array}{|c|c|c|c|} \hline & \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \\ \hline \mathbf{G}_1 & * & & \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|c|} \hline & \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \\ \hline \mathbf{G}_1 & * & & \\ \mathbf{G}_2 & * & & * \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|c|c|} \hline & \mathbf{F}_1 & \mathbf{F}_2 & \mathbf{F}_3 \\ \hline \mathbf{G}_1 & * & & \\ \mathbf{G}_2 & * & & * \\ \mathbf{G}_3 & * & & * \\ \hline \end{array} \quad (19)$$

All of the remaining candidates are evaluated in terms of the mutual information predicted to provide to other candidate measurements, and then selected based on an information efficiency score [8] 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 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:

$$I = I_{\text{discrete}} + I_{\text{continuous}}. \quad (20)$$

One of either 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.

4.2.1. 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 it is predicted to provide by

$$I(\mathbf{x}; \mathbf{z}) = H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}). \quad (21)$$

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

$$I_{\text{discrete}} = H(\mathbf{x}) - P(\mathbf{z} = \text{null}) \times H(\mathbf{x}|\mathbf{z} = \text{null}) \quad (22)$$

$$- P(\mathbf{z} = \text{match}) \times H(\mathbf{x}|\mathbf{z} = \text{match}). \quad (23)$$

where

$$H(\mathbf{x}) = \sum_{i=1}^K \lambda_i \log_2 \frac{1}{\lambda_i} \quad (24)$$

$$H(\mathbf{x}|\mathbf{z} = \text{null}) = \sum_{i=1}^K \lambda'_i \log_2 \frac{1}{\lambda'_i} \quad (25)$$

$$H(\mathbf{x}|\mathbf{z} = \text{match}) = \sum_{i=1}^{K+1} \lambda''_i \log_2 \frac{1}{\lambda''_i}. \quad (26)$$

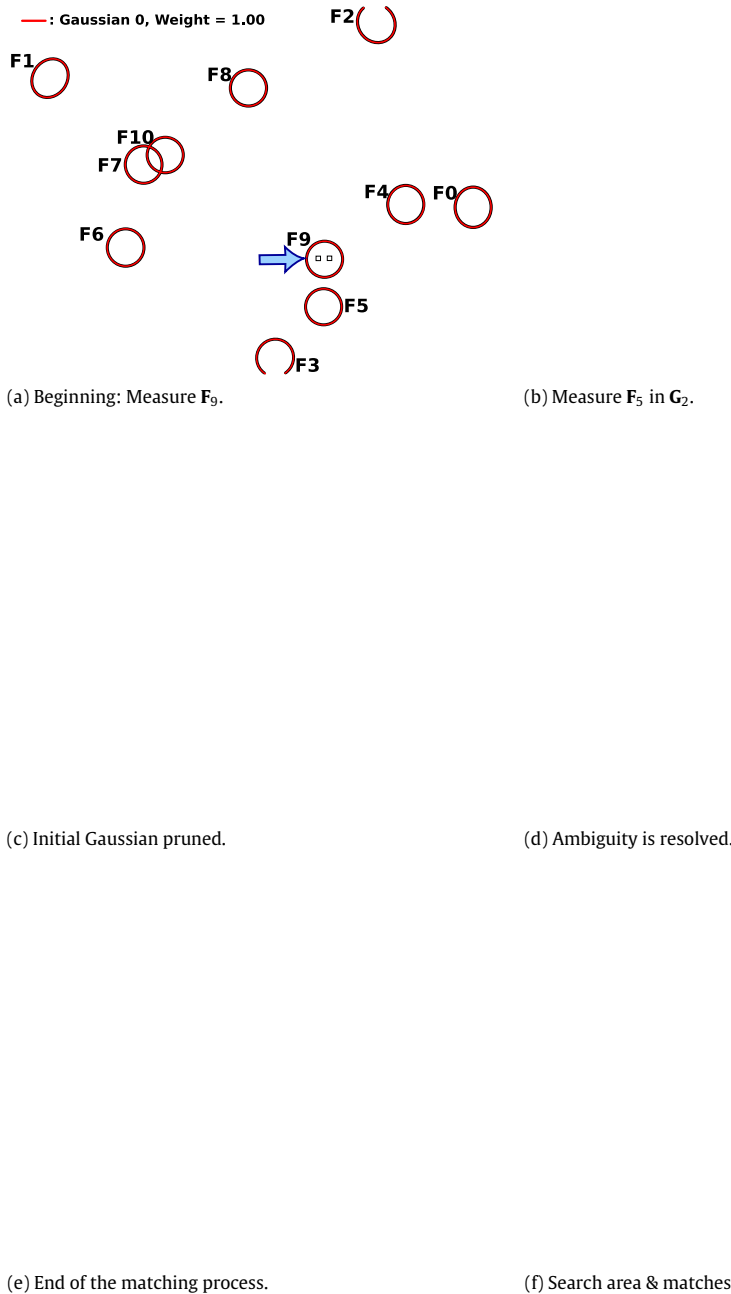


Fig. 2. Resolving ambiguity using AM. Based on the input prior on feature locations F_9 is predicted to give the most MI/(pixel searched). Propagating the outcome of (a) G_1 and G_2 are spawned in (b). The match found for F_5 in G_2 boosts the newly spawned G_3 , weakening G_0 and G_2 enough to get pruned off the mixture in (c). The match for F_{10} comes to resolve the ambiguity in (d) with G_4 having dramatically reduced width. Measuring the rest of the features, AM comes to an end in (e) and in (f) is a superposition of the area searched to achieve data association: AM searches $7\times$ less image area than standard ‘get matches first, resolve later’ approaches like JCBB.

The predicted weights after a null or a match search are calculated as in Eq. (18) 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.

4.2.2. Mutual information: Continuous component

Davison [8], building on early work by others such as Manyika [14], 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, he has shown that the mutual information between any two partitions of the state vector can be computed in absolute number of bits as in Eq. (5). Following our efficient

formulation of this expression described in Eq. (7), we compute the continuous component of the mutual information for feature f by

$$I_{\text{continuous}} = \frac{1}{2} P(z_f = \text{match}) \lambda_m'' \log_2 \frac{|P_{z_f \neq f}| |P_{z_f}|}{|P_{x_m}|}. \tag{27}$$

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 [8] but is now scaled and combined with the discrete component arising due to the expected change in the λ_i distribution.

