ARTICLE IN PRESS

Robotics and Autonomous Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

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 xxxx

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.

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).

0921-8890/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2009.07.010

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 $\mathbf{z}_T = \begin{pmatrix}\mathbf{z}_1 & \mathbf{z}_2 & \dots\end{pmatrix}^\top$ containing all candidate feature measurements and stacked likelihood function $p(\mathbf{z}_T|\mathbf{x})$, the density:

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

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. 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:

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010

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

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

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 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:

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

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

The lower-right portion of $\mathrm{P}_{\mathbf{x}_m}$ 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* S in Kalman filter tracking. The correlations between different candidate measurements mean that generally 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:

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

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

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

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

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:

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

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

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

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

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

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:

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

where we have now used the further notational shorthand $\mathbf{G}_i = \mathbf{G}(\hat{\mathbf{x}}_{m_i}, \mathrm{P}_{\mathbf{x}_{m_i}})$. 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($\mathbf{G}_0$)

```
Mixture = [[1, G_0]]   // consists of [weight, Gaussian] tuples
[f_c, G_c] = get_max_gain_candidate(Mixture)
while (pair_not_yet_measured(f_c, G_c))
    Matches = measure(f_c, G_c)
    UpdateMixture(Mixture, c, f_c, Matches)
    prune_insignificant_gaussians(Mixture)
    [f_c, G_c] = get_max_gain_candidate(Mixture)
end while
G_best = find_most_probable_gaussian(Mixture)
returnG_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  [λ_i, G_i] = Mixture[i]
2  for k = 1 : K
   // loop through all Gaussians to update them accordingly
3    if k = i then   // this is the measured Gaussian
4      for m = 1 : M   // for every match, spawn a new
       Gaussian
5        G_m = spawn_gaussian_and_fuse_match(G_i,
         Matches[m])
6        λ_m = λ_i × μ_match × prior(Matches[m], G_i)
7        Mixture = [Mixture, [λ_m, G_m]]
       end for
8      λ_i = λ_i × μ_in × (1 − prior_sum(Matches, G_i))
9      Mixture[i] = [λ_i, G_i]
     else
       // Total probability of G_k in the region covered by G_i:
10     prob= prior_sum_underG_i(G_k)
11     sum = prior_sum(Matches, G_k)
12     λ_k = λ_k × [μ_match × sum + μ_in
           × (prob − sum) + μ_out × (1 − prob)]
13     Mixture[k] = [λ_k, G_k]
     end if
14 end for
15 normalize_weights(Mixture)
16 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 $\mathbf{z}_f = \begin{pmatrix} \mathbf{z}_{f_1} & \dots & \mathbf{z}_{f_M} \end{pmatrix}^\top$ 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:

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

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

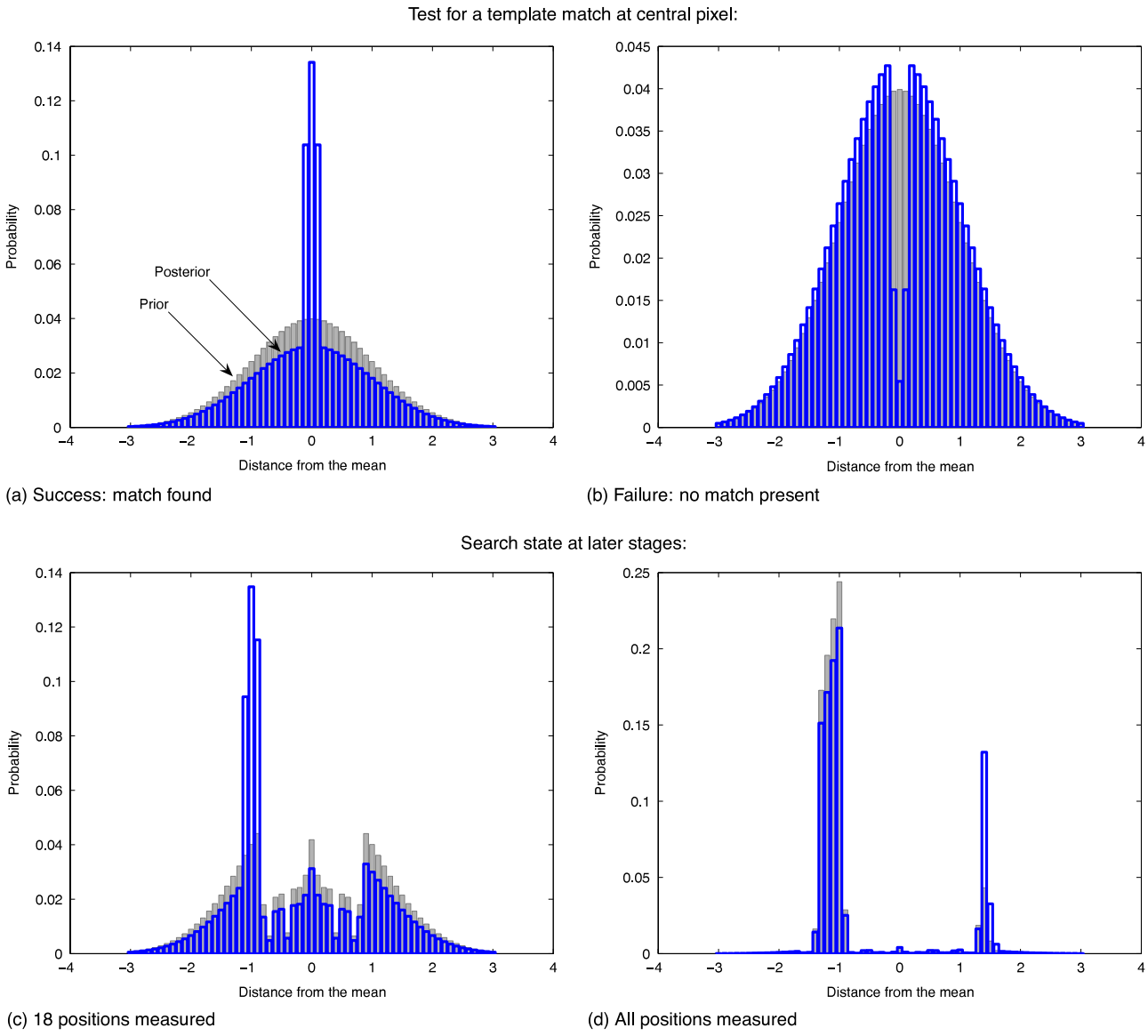


Fig. 1. One-dimensional pixel-by-pixel feature search. A full normalised histogram representing the probability that a feature is truly at each image location is refined sequentially from an initial Gaussian prior as each pixel location is tested for a template match. (a) and (b) show the outcome of 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 $\mathbb{R}_i$, and

- Two constant terms: $\mathbf{T}_{\text{in}}$ representing the hypothesis that the true match lies **in** the searched region but has not been recognised, and $\mathbf{T}_{\text{out}}$ 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:

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

$$\mu_{\text{out}} = P_{\mathbf{fp}}^{M} P_{\mathbf{tn}}^{N-M} \tag{15}$$

$$\mu_{\text{match}} = P_{\mathbf{tp}} P_{\mathbf{fp}}^{M-1} P_{\mathbf{tn}}^{N-M}, \tag{16}$$

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 this feature. $\mathbf{T}_{\text{in}}$ and $\mathbf{T}_{\text{out}}$ 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:

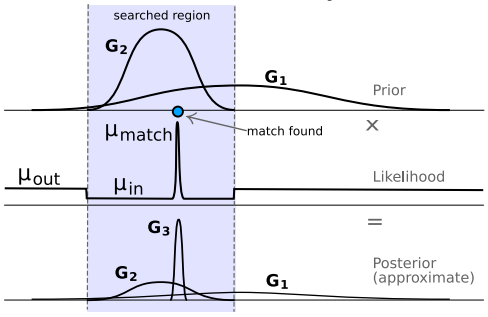
$$p(\mathbf{x}|\mathbf{z}_f) = \frac{p(\mathbf{z}_f|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z}_f)}. \tag{17}$$

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

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

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 singe 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:

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

$\Rightarrow$

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

$\Rightarrow$

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

(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:

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

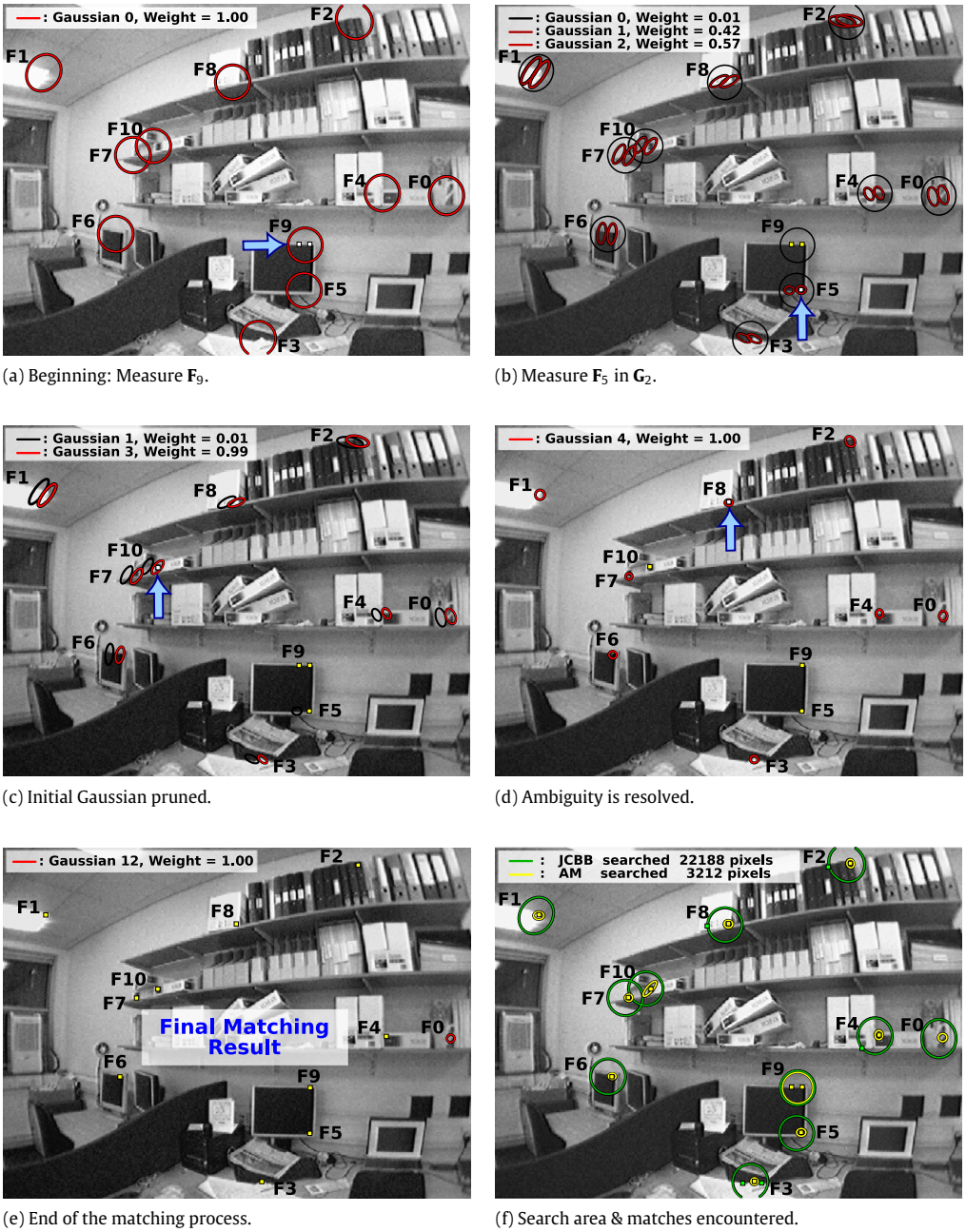
$$- P(\mathbf{z} = \text{match}) \times H(\mathbf{x}|\mathbf{z} = \text{match}). \quad (23)$$

where

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

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

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



(a) Beginning: Measure $\mathbf{F}_9$.

(b) Measure $\mathbf{F}_5$ in $\mathbf{G}_2$.

(c) Initial Gaussian pruned.

(d) Ambiguity is resolved.

(e) End of the matching process.

(f) Search area & matches encountered.

Fig. 2. Resolving ambiguity using AM. Based on the input prior on feature locations $\mathbf{F}_9$ is predicted to give the most MI/(pixel searched). Propagating the outcome of (a) $\mathbf{G}_1$ and $\mathbf{G}_2$ are spawned in (b). The match found for $\mathbf{F}_5$ in $\mathbf{G}_2$ boosts the newly spawned $\mathbf{G}_3$, weakening $\mathbf{G}_0$ and $\mathbf{G}_2$ enough to get pruned off the mixture in (c). The match for $\mathbf{F}_{10}$ comes to resolve the ambiguity in (d) with $\mathbf{G}_4$ 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× 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

$$I_{\text{continuous}} = \frac{1}{2} P(\mathbf{z}_f = \text{match}) \lambda''_m \log_2 \frac{|\mathrm{P}_{\mathbf{z}_{T\neq f}}||\mathrm{P}_{\mathbf{z}_f}|}{|\mathrm{P}_{\mathbf{x}_m}|}. \tag{27}$$

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.

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010

5. Results

We present results on the application of the algorithm to feature matching for several different situations within the publicly available MonoSLAM system [12] for real-time probabilistic structure and motion estimation. After discussing initial results in this section, we give a detailed analysis of how performance varies with different factors in Section 6 and then show how this can lead to a more efficient variant in Section 7.

MonoSLAM 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 30 Hz. 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 [7,16] it has been observed that Joint Compatibility testing finds a significant number of matching errors and greatly improves performance.

Uncertainty in the probabilistic prediction of feature image locations in MonoSLAM is dominated by uncertainty in camera pose introduced by the frame-to-frame motion model. MonoSLAM uses a constant velocity motion model which asserts that between one frame and the next the camera will experience linear and angular changes in velocity which are unknown in detail but can be probabilistically characterised by a zero-mean Gaussian distribution. The variance of the Gaussian distribution used depends on both the level of dynamic motion expected of the camera and the inter-frame time interval. Large frame-to-frame motion uncertainty occurs when vigorous, jerky movements are expected, or when the frame-rate is low. Smooth motions or high frame-rates allow more precise motion predictions and with lower uncertainty. In most cases where MonoSLAM has been applied (for example in tracking the motion of a hand-held camera in an indoor scene for use in augmented reality), in fact the angular term is dominant in the motion uncertainty's effect on image search regions since clearly it is much easier to induce fast feature motion through rotation than translation. Note that this fact has been harnessed directly in recent state of the art visual SLAM results [17] where an explicit multi-stage tracking pipeline first performs simple but effective camera rotation estimation before tracking features to estimate pose. We would hope that Active Matching would be able to exhibit similar behaviour automatically.

5.1. Algorithm characterisation

Our Active Matching algorithm simply takes as input from MonoSLAM the predicted stacked measurement vector $\mathbf{z}_T$ and innovation covariance matrix S for each image and returns a list of globally matched feature locations which are then digested by MonoSLAM's filter. Fig. 2 demonstrates Active Matching's step-by-step procedure on a typical MonoSLAM frame as it selects measurements selectively to remove ambiguity and maximise precision.

5.2. Initial sequence results

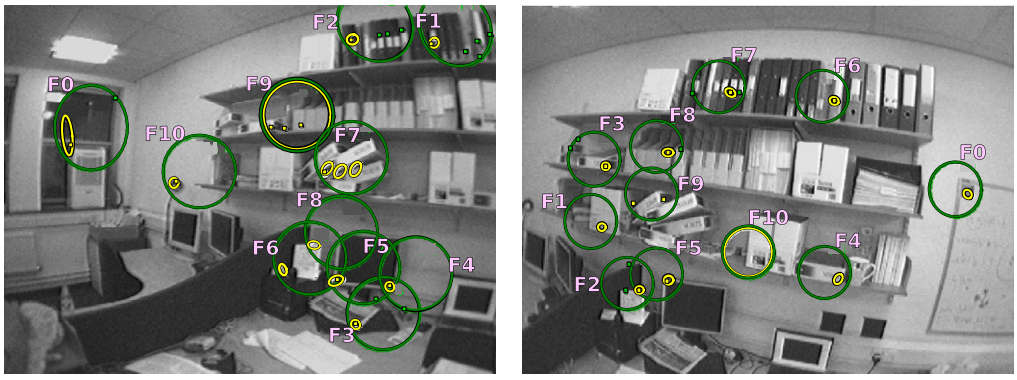
Two different hand-held camera motions were used to capture image sequences at 30 Hz: one with a standard level of dynamics slightly faster than of in the results of [12], and one with much faster, jerky motion. MonoSLAM's motion model parameters were tuned such that prediction search regions were wide 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 30 Hz sequences by a factor of two. These four sequences were all processed for 11 features per frame 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 studied. This observation confirms that the Gaussians spawned throughout the process of matching in each frame were placed around the 'correct' matches, and also that the weight-scaling of the different hypotheses has been consistent with reality; if a Gaussian had got a low weight without enough evidence of it being an unlikely scenario then it could be mistakenly pruned off the mixture resulting in missing some of the correct matches in the final, accepted result. This highlights the importance of our fully probabilistic weighting scheme but also the guidance of matching using the mutual information cues to measure the most reliable and informative features first — it would not be a sensible strategy to search for a very common feature (with a high false-positive rate) when there are more distinctive features present, or implode the weight of the searched hypothesis after a null-search of a hardly recognisable feature (low true-positive rate).

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

	One tracking step	Matching only	No. pixels searched	[relative ratio]	Max no. live Gaussians
Fast Sequence at 30 Hz (752 frames)					
JCBB	56.8 ms	51.2 ms	40341	[8.01:1]	–
AM	21.6 ms	16.1 ms	5039		7
Fast Sequence at 15 Hz (376 frames)					
JCBB	102.6 ms	97.1 ms	78675	[8.27:1]	–
AM	38.1 ms	30.4 ms	9508		10
Slow Sequence at 30 Hz (592 frames)					
JCBB	34.9 ms	28.7 ms	21517	[6.89:1]	–
AM	19.5 ms	16.1 ms	3124		5
Slow Sequence at 15 Hz (296 frames)					
JCBB	59.4 ms	52.4 ms	40548	[7.78:1]	–
AM	22.0 ms	15.6 ms	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 1 ms per frame.) Testing fewer pixels for a template match, has the immediate effect of fewer matches being encountered. Guiding the matcher to 'look' at carefully selected (reduced) regions, we avoid introducing additional confusion to the sys-

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010



(a) Fast camera motion at 15 Hz.

(b) Slow camera motion at 15 Hz.

Fig. 3. Active matching dramatically reduces image processing operations while still achieving global matching consensus. Here is a superposition of the individual gating ellipses searched in order to generate candidates for outlier rejection by JCBB(large, green ellipses) and the yellow ellipses searched for our Active Matching [9] method. In these frames, joint compatibility needed to search 8.4× more image area than active matching in (a) and 4.8× in (b). Moreover, the 'intelligent' guidance of where to search in AM, pays off in terms of the matches encountered (yellow blobs) avoiding introducing unnecessary confusion in the system with the extra matches (green blobs) encountered in JCBB. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

tem by extra false-positives improving the odds of converging to the true association scenario. The dramatic reduction in the area searched together with the matches encountered by the two techniques are overlaid on frames from two of the sequences in Fig. 3.

In all the experiments presented in this work, we have used the Shi–Tomasi criterion [18] to extract the features tracked. However, our Active Matching algorithm is not specifically tied to any particular feature detector/descriptor. While SIFT [19] or SURF [20] features would be particularly useful for matching due to their highly descriptive and distinctive nature (especially in the presence of only weak priors) the cost associated with their extraction renders them unsuitable for frame-rate matching (depending on the number of features tracked per frame). Despite the somewhat lower quality alternatives like Shi–Tomasi, FAST [21,22] features or the randomised ferns classifier [23] as used in [16], these could be used equally effectively in matching — allowing denser frame-to-frame correspondence scenarios studied in the next section.

6. Detailed performance analysis

In order to assess the performance of Active Matching in detail, we have generated a set of experimental sequences by taking a high frame-rate image sequence and down-sampling temporally to generate reduced versions. Varying both the frame-rate and the number of features being tracked per frame, we generate a matrix of experiments to form the testbed of performance assessment of Active Matching.

6.1. Performance with varying frame-rate and number of features

In this analysis of the computational performance of Active Matching, we consider the average time consumed per frame in terms of the main stages of the algorithm. Namely, within each matching step it is necessary to (i) **evaluate** the mutual information that each candidate measurement is predicted to provide followed by (ii) **measurement** of the selected candidate (by correlation) and finally (iii) the **update** of the mixture of Gaussians according to the measurement result.

For the sake of comparison with the 'get candidates first, resolve later' methods, we monitor the computational time needed to perform JCBB. Again, the timings are considered in terms of the time consumed to perform the two main steps of the method, namely to (i) **get the candidate matches** for each feature (by correlation) and (ii) **resolve** their consensus.

6.1.1. Fixed frame-rate; varying number of features

Increasing the number of features tracked per frame means that the matcher is equipped with more evidence to aid the resolution of ambiguities, and in general it has been shown that tracking many features is key in obtaining more precision in pose estimation [17] and therefore is clearly desirable. On the other hand, more time needs to be consumed to process the extra information available. In order to study how much more time is needed we recorded timings while varying the number of features matched per frame when tracking a particular sequence. Time breakdowns for both Active Matching and Joint Compatibility are shown in Fig. 5.

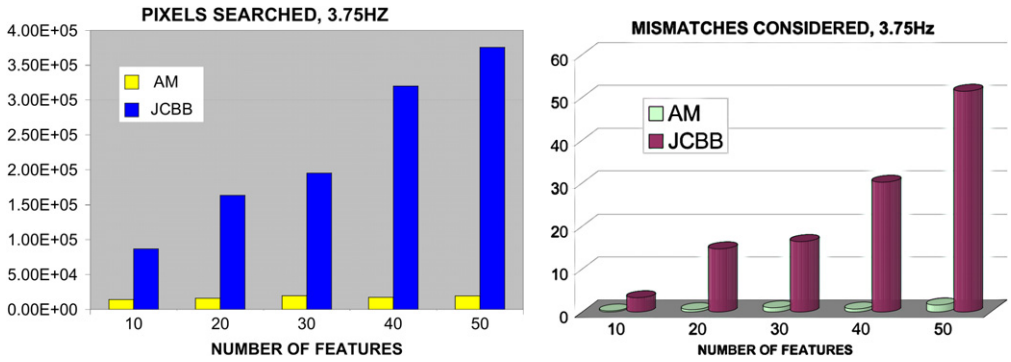
Our results show that Active Matching scales badly with increasing number of features and the step dominating the time consumed is the mutual information calculation of the candidate measurements in order to select which one to measure next. This is explained by the fact that every new feature added in the system, introduces a new candidate measurement for **each** Gaussian present in the mixture. Therefore, Active Matching has more candidates to choose from, especially in a highly ambiguous scene where there are many Gaussians present (i.e. in the low frame-rate case in Fig. 5(a)). Evaluating the MI of each candidate, involves a prediction of how the MoG will evolve in both cases of a successful and a failed measurement of the current candidate. The estimation of the continuous MI part in particular, translates into the potentially costly handling of big Innovation Covariance matrices — which expand linearly with the number of features.

Joint Compatibility performs better with increasing number of features, but is still far from real-time performance. Measuring more features translates into more image regions we need to search for template matches but also potentially more false-positives — hence the constantly increasing time needed to perform correlation and resolve consensus. Active Matching on the other hand, since it is being very selective in the areas it looks for matches, both of the number of mismatches encountered as well as the number of pixels searched remain very low even for big numbers of features matched as demonstrated in Fig. 4.

6.1.2. Coping with ambiguity: Varying frame-rate; fixed number of features

As the frame rate decreases and the search-regions of features cover bigger image area, it becomes more likely to encounter more mismatches per feature, therefore complicating the process of discovering the consensus in the prediction error. This is evident

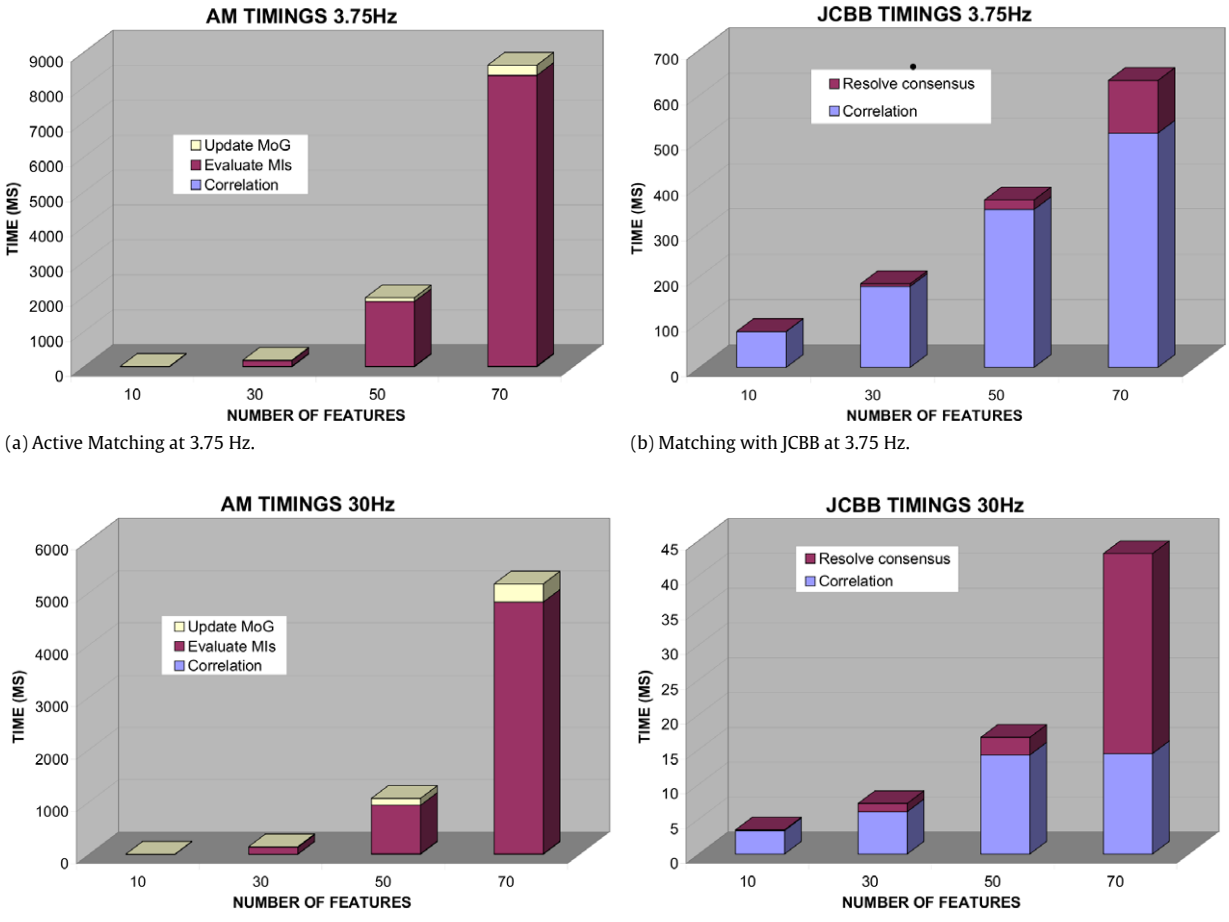
Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010



(a) Pixels searched per frame.

(b) Mismatches encountered per frame.

Fig. 4. Carefully selecting where to look for matches pays off for Active Matching which needs to search dramatically fewer pixels per frame than JCBB as demonstrated in (a). Also, constantly refining the search region for each feature avoids encountering unnecessary false positives, which is the case with Joint Compatibility as shown in (b).



(a) Active Matching at 3.75 Hz.

(b) Matching with JCBB at 3.75 Hz.

(c) Active Matching at 30 Hz.

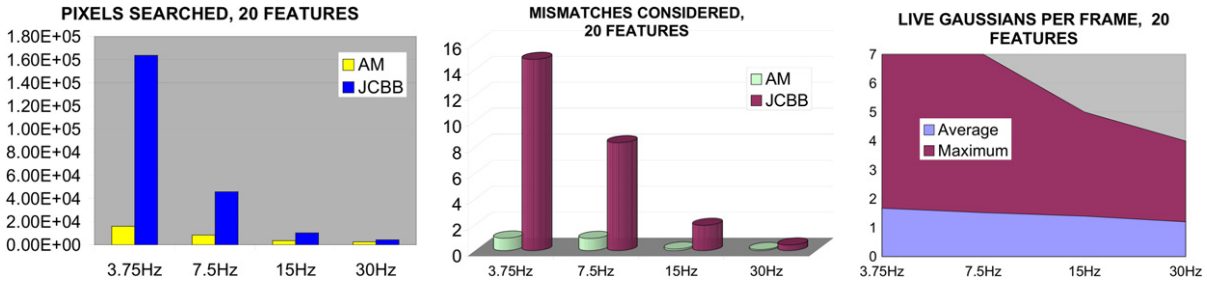
(d) Matching with JCBB at 30 Hz.

Fig. 5. Computational time breakdown for AM and JCBB while varying the number of features matched in the 3.75 Hz (top row) and at 30 Hz (bottom row) sequences. Active Matching scales badly with increasing number of features mainly due to the constantly expanding cost of the evaluation step of the mutual informations of all the measurement candidates. Joint compatibility on the other hand, maintains better performance when more candidate measurements are available but its performance is also far from real-time due to the increasing number of pixels needed to test for a template match.

in Fig. 6 where again, the number pixels searched is dramatically reduced using Active Matching and as a result so is the number of mismatches encountered. As matching becomes more ambiguous with decreasing frame rate, we need more Gaussians in the mixture to accurately represent the different hypotheses arising, hence the negative slope in the maximum and average number of live Gaussians in Fig. 6(c).

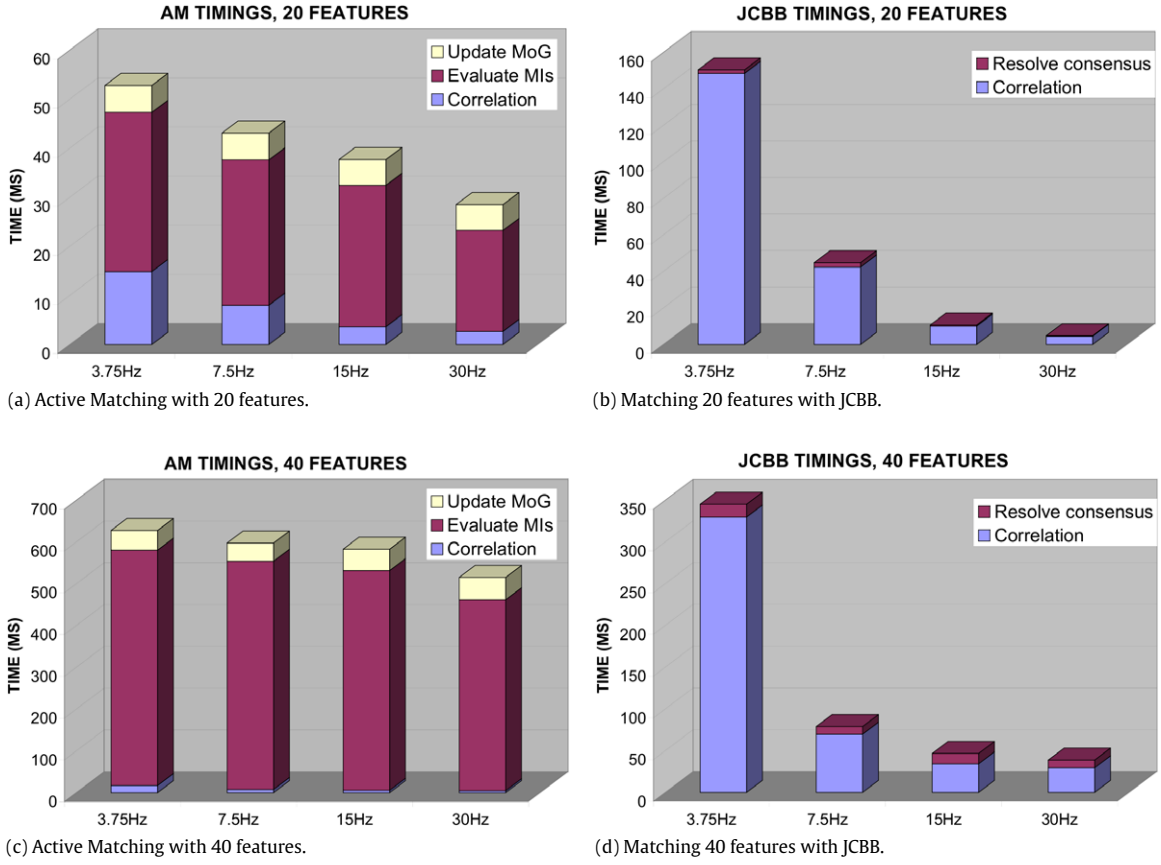
Tracking a scene with a low frame-rate camera is the real challenge for data association algorithms since the amount of time elapsing between consecutive frames is increasing, introducing larger uncertainty into the system. The uncertainty in the camera position translates into inflated search regions for each feature in the image plane. In this set of experiments we aimed to assess the performance of Active Matching in the presence of high ambiguity,

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010



(a) Pixels searched. (b) Mismatches encountered. (c) Live Gaussians in mixture.

Fig. 6. Decreasing the frame rate more pixels need to be tested for a match as shown in (a). This also means that more ambiguity is present during matching as more mismatches are likely to occur as demonstrated in (b). When tracking highly ambiguous sequences, more matching scenarios arise per frame, hence the mixture of Gaussians needs to be populated with more members as confirmed in (c), in order to accurately reflect the search-state at every instant.



(a) Active Matching with 20 features. (b) Matching 20 features with JCBB.

(c) Active Matching with 40 features. (d) Matching 40 features with JCBB.

Fig. 7. Timings breakdown for variable frame rate matching a constant number of features using Active Matching and JCBB (tracking 20 features per frame in the top row and 40 in the bottom row). For around 20 features per frame, Active Matching is entirely within real-time limits for all frame-rates whereas JCBB's performance degrades at low frame-rates since more time is needed to find the correlation matches. When tracking 40 features per frame though, the costly evaluation of MIs pushes the time performance of Active Matching lower.

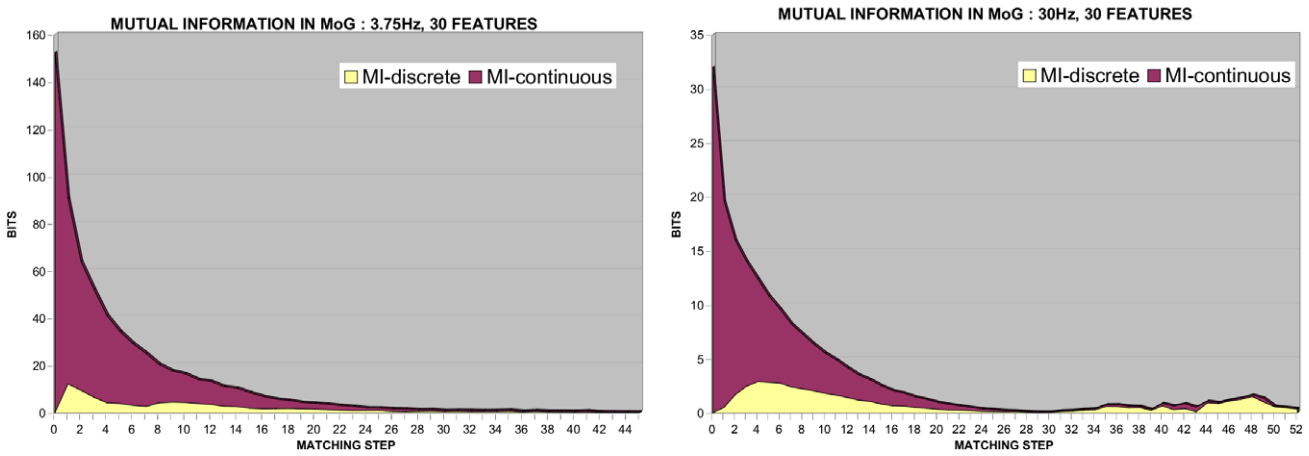
by tracking four consecutively subsampled sequences, with the initial one grabbed at 30 Hz (keeping the number of tracked features per frame constant). The breakdown of timings is shown in Fig. 7.

6.2. Evolution of mutual information

Mutual information is what guides our matcher to select potentially more informative measurements, avoiding areas of high ambiguity. Since the process of evaluating the discrete and continuous parts for every candidate has been proven to be the main computational bottleneck of our algorithm, here we study the evolution of the mutual information throughout the matching steps of each frame to uncover the true value it has at different stages during matching.

As demonstrated in Fig. 8 at the beginning of matching there is no ambiguity in the mixture since we start off with one Gaussian with high uncertainty (which is directly related to the frame-rate of tracking). This is represented by the dominant MI-continuous presence during the initial steps of matching, since this part of MI takes account of the desire to improve the accuracy of the most probable Gaussian. As we obtain matches for more features, the MI-continuous decreases dramatically and if any of the matches encountered is inconsistent with existing Gaussians, new ones

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010



(a) MI components at 3.75 Hz.

(b) MI components at 30 Hz.

Fig. 8. Evolution of the continuous and discrete components of MI for different frame rates, throughout the matching steps followed during AM in an average frame. In both subfigures the two MI parts are shown stacked on top of each other to demonstrate the contribution that each has to the total MI in the mixture at any given step. The Continuous-MI is the dominant factor during the initial steps of matching, especially when tracking at 3.75 Hz in (a) where there is more uncertainty present. As features get localised one-by-one, the uncertainty in the MoG decreases, but as soon as we start encountering inconsistent measurements, more Gaussians are spawned resulting to an increase in the Discrete-MI part which aims at resolving ambiguity. In both (a) and (b), the total MI tails off smoothly (notice the difference in scale) as the matcher encounters more measurements.

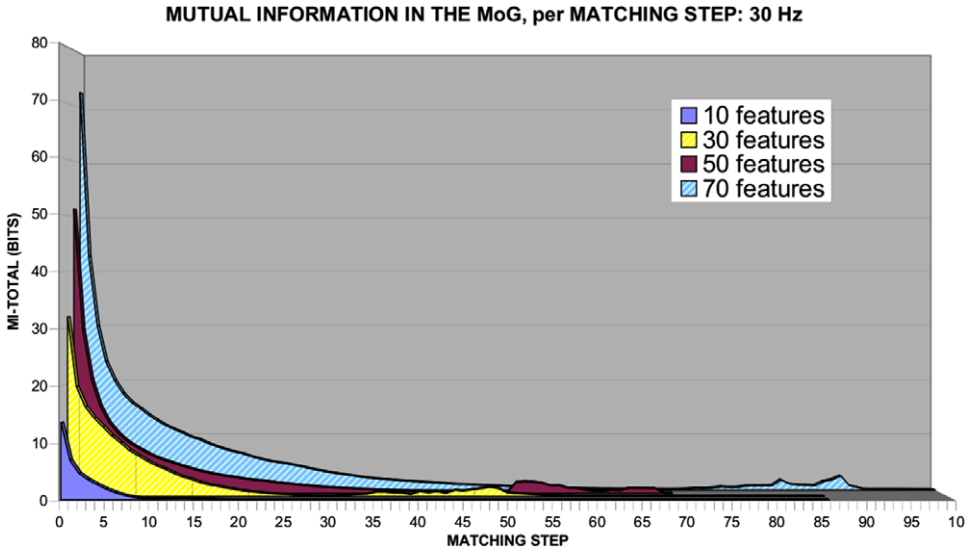


Fig. 9. Matching many features is informative. But how much more information is a new measurement expected to give? This figure shows that the more the features we match per frame, the more information we expect to get during the initial steps of matching. After matching has progressed for a number of steps though, the MI present in the mixture does not decrease significantly.

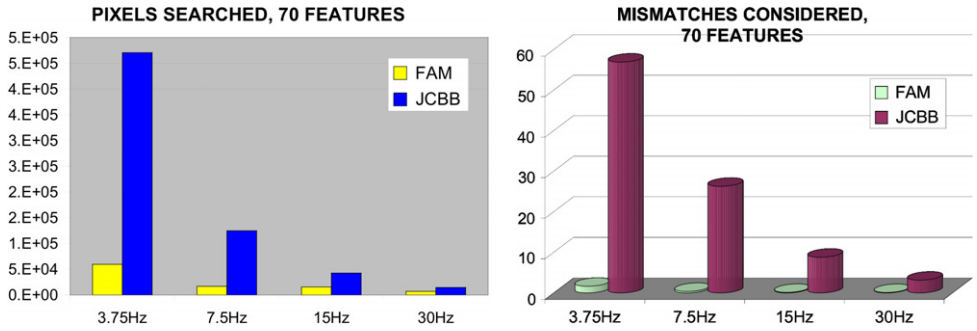
are spawned to accurately reflect the ambiguous search-state. In such cases, the MI-discrete part comes in and sometimes takes over until both resolution of ambiguity and high accuracy are achieved.

The more features we match, the more information we expect to gain, always at the expense of computational time. So is it really worth the effort measuring one more feature? How much more information lies in this candidate measurement? A good answer to this question relies on a plethora of factors; feature characteristics, camera dynamics, speed of processor, etc. The evolution of the total mutual information in the mixture can be a representative measure of the value that an extra measurement can have in the current search-state. Fig. 9 demonstrates that despite that initially there is higher mutual information to be gained for a bigger numbers of features, as we proceed with matching features one-by-one the total-MI decays exponentially. During the initial steps of the process, the evaluation of predicted MIs is key to the algorithm since most of the uncertainty and ambiguity in the scene get resolved. Measuring an extra feature after a certain stage though does not really tell much more to the current search state. Thus, predicting which feature will provide the most information to measure next does not have any significant effect to the subsequent result of the algorithm. These observations and conclusions are exploited below to refine our Active Matching method so that it can dynamically adapt its performance according to the number of features and ambiguity in tracking, achieving improved computational performance without compromising accuracy.

7. Fast Active Matching

We have seen that Active Matching is being very selective in the areas it looks for matches of features and this really pays off in terms of the number of mismatches encountered and hence aids the resolution of ambiguity. On the other hand, the process of evaluating *all* the {Feature, Gaussian} candidate measurement com-

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010



(a) Pixels searched per frame.

(b) Mismatches encountered per frame.

Fig. 10. Comparing Fast Active Matching (FAM) with JCBB. Correlation is now the dominant factor in both methods, but in FAM the pixels searched are an order of magnitude less in some cases as demonstrated in (a), explaining the superior performance of the algorithm. Subfigure (b) shows the difference in mismatches encountered by the two methods which is the cause of inflated timings to resolve consensus in JCBB.

binations is the main bottleneck in computational performance. Since we have discovered that only the first steps of matching are the crucial parts in decreasing variance and resolving ambiguity, we propose to stop evaluating mutual informations once the total MI has dropped below a threshold. As demonstrated in Fig. 9 where the total MI is shown to tail off relatively early during matching, this 'approximation' has a negligible effect on the course of the algorithm, as expected.

Despite that aborting evaluation of MIs after a certain stage will have a big impact in the computation time, if we want to track many features we will still have to deal with big matrices, primarily during the evaluation of MIs (for the first steps) but also during the update of the mixture alone. Therefore, we propose to cut down the computational cost by pre-selecting a certain number of candidates to evaluate their MIs rather than evaluating all of them. It is most likely that we no longer will be able to discover the *best* candidate to measure next, but provided that the pre-selected candidates are evenly spread across all Gaussians (one can easily select a certain number of candidates from each Gaussian), the candidate that gets selected for measurement should be a fairly good approximation. Within each Gaussian, the pre-selection is random.

Once enough measurements have been made and the total MI of the mixture has dropped enough to stop evaluating MIs, we can check if there is a dominating Gaussian with high enough probability. In case there is such a dominating Gaussian, under these MI conditions it will have very low uncertainty left (since there is no ambiguity to push MI-discrete scores up) so fusing the nearest neighbour matches for the yet-unmeasured features is guaranteed to produce the most consistent scenario.

In Figs. 11 and 10 we demonstrate how these refinements can dramatically improve the computation time of Active Matching to the extend that it beats JCBB. All the results shown in this section have been taken by pre-selecting 15 random candidates evenly spread across all Gaussians. Evaluation of MIs stops when the total-MI per feature drops below 0.5 bits and if also there is a dominating Gaussian with more than 70% probability of being correct we accept it as the true scenario, fusing the nearest neighbour matches to the remaining unmeasured features. Note that since we prune weak Gaussians throughout matching and renormalise the weights, a Gaussian with probability 70% by the end of the matching is actually a lot more certain.

In the future, we can go a step even further to stop measuring features when the MI in the mixture becomes very low. This is expected make use of the fully adaptive nature of Active Matching and can prove particularly beneficial in high-frame rate tracking with a lot of features. In such cases, the uncertainty in the camera pose can be very small leaving little room for ambiguities during matching. Also, the expected improvement in accuracy with more measurements can soon be ruled insignificant therefore, aborting matching at that stage translates into reducing redundancy with potentially big savings in computation time.

8. Conclusions

This work demonstrates how a mixture of Gaussians formulations allow 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.

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. The priors on absolute feature locations will be weak but priors on relative locations may still be strong.

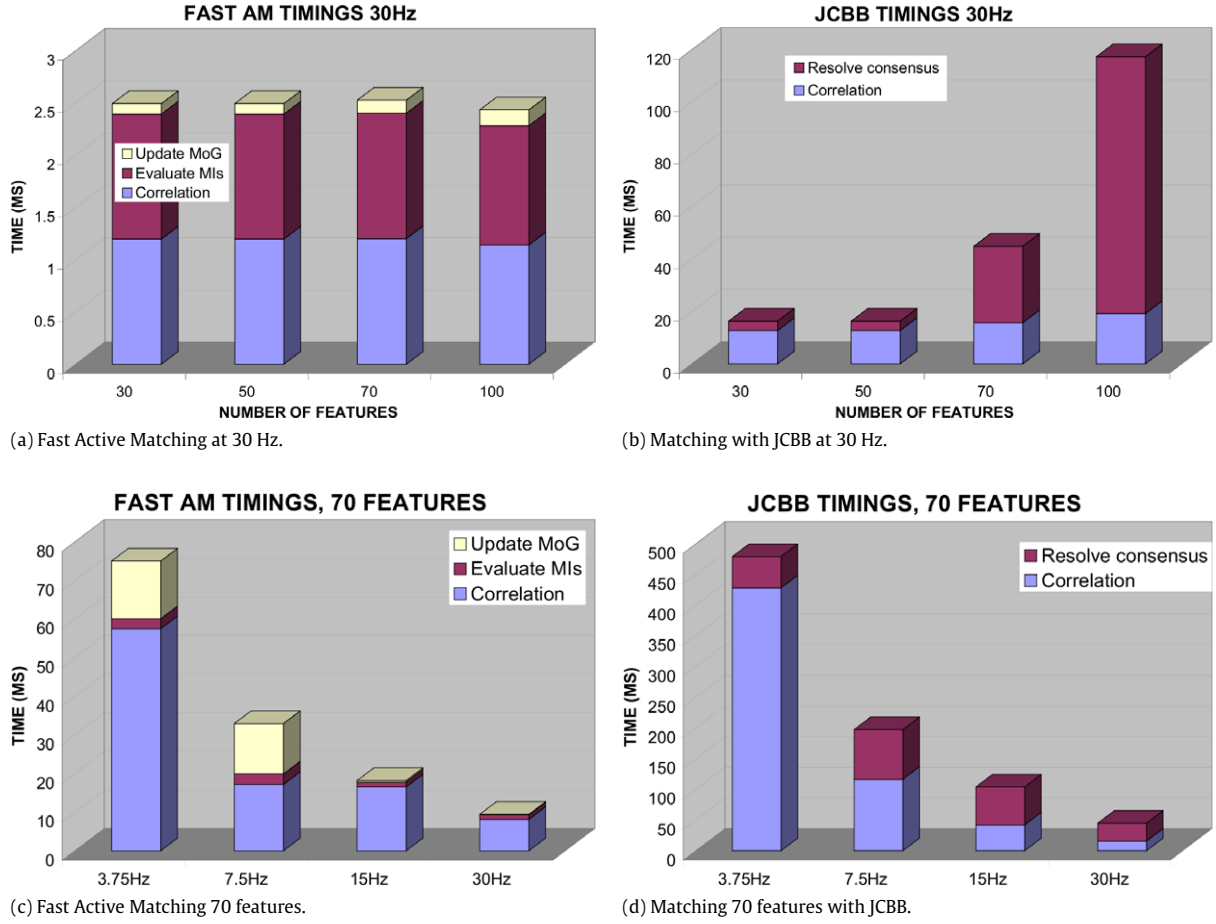
In this article, we presented an evaluation of the performance of Active Matching via extensive testing for variable number of features tracked per frame and different frame-rates, in an attempt to unveil the bottlenecks of the algorithm in comparison to standard 'get candidates first, resolve later' approches like JCBB. Briefly, our results indicate that the full Active Matching algorithm despite maintaining real-time performance for different frame-rates for a relatively low number of features per frame (around 20), it scales badly when this number increases mainly due to the manipulation of large matrices during the calculation of mutual information.

Following a detailed discussion of the value of mutual information in the course of the algorithm, we observed that carefully selecting which feature to measure at each step (guided by mutual information) plays a key role during the initial steps of matching where most of the uncertainty and ambiguity in the systems gets resolved. Making use of the fact that in later stages of the algorithm the search state usually converges to a single, dominant hypothesis, we present our Fast Active Matching algorithm which achieves real-time performance for large numbers of features even when JCBB does not, through some minor approximations.

In future work, we aim to look into techniques to track even more features, faster. We believe that mutual information has yet a lot to provide in high frame-rate tracking — the motion priors are indeed stronger then but the limited processing time available makes the task of resource allocation in matching even more challenging.

Our long-term aim is to develop fully scalable algorithms via the active matching approach which will be able to perform the best matching job possible given a certain computational budget. For instance, state of the art optical flow algorithms [24] are now able

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010



(a) Fast Active Matching at 30 Hz.

(b) Matching with JCBB at 30 Hz.

(c) Fast Active Matching 70 features.

(d) Matching 70 features with JCBB.

Fig. 11. Superior performance of Fast Active Matching over JCBB. Subfigures (a) and (b) show the computational time breakdown for fast AM and JCBB respectively when tracking at 30 Hz; the time spent in evaluation of MIs here is significantly reduced maintaining almost constant overall time adapting to the number of features whereas the resolution of consensus in JCBB deteriorates performance with increasing number of features. In (c) and (d) are the timings for tracking 70 features at different frame rates.

to produce real-time matching for every pixel in an image when running on the latest GPU hardware. A hierarchical active approach may permit such dense matching performance to be approached with much reduced computational requirements.

Acknowledgements

This research was supported by EPSRC grant GR/T24685/01 and ERC Starting Grant 210346. We are grateful to Ian Reid, José María Montiel, José Neira, Javier Civera, Paul Newman and Peter Gemeiner for very useful discussions.

References

[1] M.A. Fischler, 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) (1981) 381–395.
[2] O. Chum, J. Matas, Optimal randomized RANSAC, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (8) (2008) 1472–1482.
[3] D. Nistér, Preemptive RANSAC for live structure and motion estimation, in: Proceedings of the 9th International Conference on Computer Vision, ICCV, Nice, 2003.
[4] B. Tordoff, D. Murray, Guided-MLESAC: Faster image transform estimation by using matching priors, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 27 (10) (2005) 1523–1535.
[5] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Transactions on Robotics and Automation 17 (6) (2001) 890–897.
[6] W.E.L. Grimson, Object Recognition by Computer: The Role of Geometric Constraints, MIT Press, Cambridge, MA, 1990.
[7] L.A. Clemente, A.J. Davison, I.D. Reid, J. Neira, J.D. Tardós, Mapping large loops with a single hand-held camera, in: Proceedings of Robotics: Science and Systems, RSS, 2007.
[8] A.J. Davison, Active search for real-time vision, in: Proceedings of the International Conference on Computer Vision, ICCV, 2005.
[9] M. Chli, A.J. Davison, Active matching, in: Proceedings of the European Conference on Computer Vision, ECCV, 2008.
[10] M. Chli, A.J. Davison, Efficient data association in images using active matching, in: Robotics: Science and Systems Workshop on Inside Data Association, 2008.
[11] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, in: Proceedings of the 4th European Conference on Computer Vision, ECCV, Cambridge, 1996, pp. 343–356.
[12] A.J. Davison, N.D. Molton, I.D. Reid, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29 (6) (2007) 1052–1067.
[13] A.J. Davison, D.W. Murray, Mobile robot localisation using active vision, in: Proceedings of the European Conference on Computer Vision, ECCV, 1998.
[14] J. Manyika, An information-theoretic approach to data fusion and sensor management, Ph.D. Thesis, University of Oxford, 1993.
[15] D. Mackay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.
[16] B. Williams, G. Klein, I. Reid, Real-time SLAM relocalisation, in: Proceedings of the International Conference on Computer Vision, ICCV, 2007.
[17] G. Klein, D.W. Murray, Improving the agility of keyframe-based slam, in: Proceedings of the European Conference on Computer Vision, ECCV, 2008.
[18] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 1994, pp. 593–600.
[19] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision (IJCV) 60 (2) (2004) 91–110.
[20] H. Bay, T. Tuytelaars, L.V. Gool, SURF: Speeded up robust features, in: Proceedings of the European Conference on Computer Vision, ECCV, 2006.
[21] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, in: Proceedings of the 10th International Conference on Computer Vision, ICCV, Beijing, 2005.
[22] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: Proceedings of the European Conference on Computer Vision, ECCV, 2006, pp. 430–443.

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010

[23] V. Lepetit, P. Fua, Keypoint recognition using randomized trees, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 28 (9) (2006) 1465–1479.
[24] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime TV-L1 optical flow, in: Proceedings of the DAGM Symposium on Pattern Recognition, 2007.

Margarita Chli has graduated with the B.A. and M.Eng. degrees in Information and Computing Engineering from the University of Cambridge, UK in 2004 and 2005, respectively. Currently, she is a candidate for the Ph.D. degree in computer vision at the Visual Information Processing group of Imperial College London, London, UK. Her research focus is the application of information theoretic techniques to achieve robust and real-time simultaneous localisation and mapping (SLAM) using vision.

Andrew J. Davison received the B.A. degree in physics and the D.Phil. degree in computer vision from the University of Oxford, Oxford, UK, in 1994 and 1998, respectively. He was with Oxford's Robotics Research Group, where he developed one of the first robot simultaneous localization and mapping (SLAM) systems using vision. He was a European Union (EU) Science and Technology Fellow at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan, for two years, where he was engaged in visual robot navigation. In 2000, he returned to the University of Oxford as a Postdoctoral Researcher. In 2005, he joined Imperial College London, London, UK, where he currently holds the position of Reader with the Department of Computing. His current research interests include advancing the basic technology of real-time localization and mapping using vision while collaborating to apply these techniques in robotics and related areas. Dr. Davison was awarded a five-year Engineering and Physical Sciences Research Council (EPSRC) Advanced Research Fellowship in 2002. In 2008, he was awarded a European Research Council (ERC) Starting Grant.

Please cite this article in press as: M. Chli, A.J. Davison, Active matching for visual tracking, Robotics and Autonomous Systems (2009), doi:10.1016/j.robot.2009.07.010