Inexact Bayesian point pattern matching for linear transformations

J. Christmas[a,*], R.M. Everson[a], J. Bell[b], C.P.Winlove[c]

[a]*Department of Computer Science, University of Exeter, Exeter, EX4 4QF, UK*
[b]*Peninsula College of Medicine and Dentistry, University of Exeter, Exeter, EX1 2LU, UK*
[c]*Department of Biophysics, University of Exeter, Exeter, EX4 4QF, UK*

Abstract

We introduce a novel Bayesian inexact point pattern matching model that assumes that a linear transformation relates the two sets of points. The matching problem is inexact due to the lack of one-to-one correspondence between the point sets and the presence of noise. The algorithm is itself inexact; we use variational Bayesian approximation to estimate the posterior distributions in the face of a problematic evidence term. The method turns out to be similar in structure to the iterative closest point algorithm.

Keywords: Bayesian methods, variational approximation, point pattern matching, iterative closest point algorithm, linear transformation

1. Introduction

Point pattern matching (also referred to as point set matching or point set registration) is a common pattern recognition problem that arises in many different fields, but perhaps particularly from the increasing use of automatic image processing techniques (e.g. [1, 2, 3, 4]). A set of feature points is extracted from each of two similar images (possibly two frames of a video) and the aim is to determine correspondences between the two sets.

*Corresponding author
Email addresses: J.T.Christmas@exeter.ac.uk (J. Christmas), R.M.Everson@exeter.ac.uk (R.M. Everson), J.S.Bell@exeter.ac.uk (J. Bell), C.P.Winlove@exeter.ac.uk (C.P.Winlove)

Often it is assumed that the two sets are related through some linear transformation and any deviations from that are regarded as noise.

We denote the two sets of points as $\mathbf{Y} = \{\mathbf{y}_i\}$ and $\mathbf{X} = \{\mathbf{x}_j\}$, where each point is represented by its location in D-dimensional Euclidean space. The points do not have identities, or at least the identities are not known, and the sets of points are unordered, that is $\mathbf{y}_n$ does not necessarily correspond to $\mathbf{x}_n$ for any n.

In the simplest, *exact* case, each point set contains the same number of points and there is an exact one-to-one correspondence between them, with no noise. Thus we have the case that $\mathbf{Y} = f(\mathbf{X})$, where the function $f(\cdot)$ permutes the points in $\mathbf{X}$ and linearly transforms their coordinates so that they precisely coincide with the points in $\mathbf{Y}$. However, the nature of real problems and the automated processes by which features are often extracted, often result in the *inexact* case, where the point sets do not exactly correspond, both because of noise and because each set contains points with no counterpart in the other. In this case $\mathbf{Y}$ and $\mathbf{X}$ may contain different numbers of points and, with $\mathbf{Y}_s$ as a subset of the points in $\mathbf{Y}$ and $\mathbf{X}_s$ as a same-sized subset of the points in $\mathbf{X}$, we have $\mathbf{Y}_s = f(\mathbf{X}_s) + \textit{noise}$. We refer to the points in $\mathbf{Y}_s$ and their counterparts in $\mathbf{X}_s$ as the *overlap* between the two sets.

The inexact problem has been shown to be NP-complete [5], that is the computation time required to find the global optimum increases exponentially with the number of points. Many methods therefore (including the one described in this paper) aim to find local optima in more acceptable time-frames.

Bottom-up approaches to this problem search directly for plausible point matches. In tree search algorithms with backtracking, for example, a partial (initially empty) set of mappings is progressively augmented with new mappings until a constraint is violated. The algorithm then backtracks, i.e. removes mappings, until some alternative route is available. Conte et al. [6] provide a useful overview of a number of important algorithms. In contrast, the top-down approach aims to determine the geometric transformation which relates the two point sets and uses that to find the point mappings. The iterative closest point algorithm (ICP) [1, 7], for example, starts with an initial estimation of the point mappings, from which it estimates the parameters of a rigid transformation (rotation and translation) using a least squares method. The set of point mappings is recalculated based on this new estimate of the transformation. The process repeats the transformation-estimation and point-mapping steps iteratively until convergence.

We introduce an approximate Bayesian model for inexact point pattern matching which, due to the necessity of avoiding a problematic likelihood term, turns out to be similar in structure to ICP. We assume that the two point sets are related by a linear transformation and explicitly model each of its parameters, and the noise, as random variables. This allows us to incorporate prior knowledge about the transformation and provides estimates of the confidence intervals in the posterior distributions for each variable. We also end up with probabilities associated with every potential match; these provide a principled method for determining both the point mappings and which points in each set are unmatched. As with many Bayesian models, the integrals required for exact inference are intractable and so we use a variational approximation method [8, 9, 10, 11]. This method minimises the Kullback-Leibler divergence [12, 13] between the approximate and actual posterior distributions to determine the optimal hyperparameter values for the approximations. Interdependencies between the expressions for the posterior parameters in the variational scheme lead to an iterative update procedure which naturally results in an ICP-like update-remap process.

We start by examining different approaches to probabilistic modelling in inexact point pattern matching and the more generalised problem of graph matching, and Bayesian approximation. The new model is described in section 2 in terms of 3-dimensional point sets, though it is easily extended to lower or higher dimensionalities. Section 3 describes the results obtained from synthetic data and in section 4.2 the method is demonstrated on a real problem of matching cartilage cells in image stacks captured before and after a stretch is applied to the cartilage and the position and orientation of the sample in the microscope's viewing window is changed.

1.1. Probabilistic approaches in point set matching

Many inexact matching algorithms relax the tight constraints imposed on exact matching by calculating a cost associated with that relaxation; the larger the deviation the higher the cost and hence the aim is to minimise the total cost. Cost calculations often explicitly define different types of constraint violation and specify heuristically established costs with each of them. The tree search Attributed Relational Graphs algorithm [14], for example, bases the cost on graph edit operations of node and edge substitution.

An intuitive alternative to cost minimisation is probability maximisation. Continuous optimisation approaches to point pattern matching start with an initial guess at the mappings which is then refined over successive iterations. One such method is relaxation labelling [15, 16], where each point in one set is assigned a vector containing the probabilities that the point is mapped to each of the points in the other set. These probabilities are initialised heuristically and then refined by taking into account the probabilities associated with adjacent points. At the end the maximum probability mapping is selected.

Relaxation labelling only enforces one-to-one correspondence in one direction. Weighted Graph Matching (e.g. [17, 18]) is a quadratic optimisation method that allows two-way enforcement by way of a matching matrix of probabilities. The graduated assignment graph matching algorithm [19] gradually increases the constraints on the matching matrix to avoid poor local optima.

Although these models use probability measures, they might not be considered to be probabilistic models. A number of different probabilistic modelling approaches have been considered, using iterative expectation maximisation (EM) algorithms to find maximum likelihood solutions. Luo and Hancock [20] consider one set of points to be latent variables and the other to be observations, casting the problem as a Markov random field. Granger and Pennec [21] define a probabilistic ICP model based on a rigid transformation and a binary matching matrix, which is considered to be a latent variable. They use an annealing scheme to improve the reliability with which the global optimum is found. Jian and Vemuri [22, 23] and Myronenko and Song [24] represent the two sets of points as Gaussian mixture models and maximise the likelihood of the point mappings. Xiao et al. [25] use a hidden Markov model to model the distribution of points in each of the sets and minimise the dissimilarity between the two models by minimising the Kullback-Leibler divergence between them. Serradell et al. [4] use a tree search algorithm with backtracking to learn an affine transformation that approximately aligns the two point sets as a starting point for modelling the localised perturbations as Gaussian Processes. The update of the affine transformation estimate is performed using a process similar to the Kalman filter.

A fully Bayesian technique avoids the pitfalls associated with the maximum likelihood method: integrating (averaging) over all possible values of the parameter variables guards against overfitting and posterior probability distributions (rather than point estimates) are calculated for each of them, from which we obtain a measure of confidence in the inference.

Zhu et al. [26] note that although ICP has been widely used for problems where the transformation is rigid, it does not work well if the transformation is, for example, affine. Du et al. [27] incorporate the affine transformation into ICP and use an iterative quadratic programming method to converge on a local optimum. They decompose the transformation matrix into three using singular value decomposition and then constrain these matrices to try and avoid the problem that the most likely transformation maps all of the points in one set onto a very small subset (often a single point) of the other set. Zhu et al. [26] avoid this problem by defining the mappings bidirectionally.

We represent each of the parameters of the linear transformation as separate random variables and use prior probability distributions to constrain them, both so that we may incorporate our prior knowledge about the likely transformation and to avoid the degenerate case described above. Point mappings are derived from a matching matrix containing probabilities for all possible mappings and from this we may also estimate which points in each set are unmapped.

As is often the case, calculation of the evidence or marginal likelihood for our Bayesian model is intractable, so we must resort to some approximation scheme. With a large number of variables, numerical methods, such as quadrature [28], are not feasible and sampling methods such as the Markov chain Monte Carlo algorithms of Metropolis-Hastings [29, 30] and Gibbs sampling [31] (e.g., [2]) or particle filtering [32] (e.g., [3]) are too computationally expensive. Instead we estimate the posterior distributions using variational Bayesian approximation, which we describe in section 2.2.

2. The model

We describe the model here in terms of 3-dimensional space; it is easily extended to spaces of lower or higher dimensionality.

Without loss of generality let $\mathbf{Y}$ be the smaller of the two sets and each set be independently mean-centred such that

$$\sum_{i=1}^{N_y} \mathbf{y}_i = \sum_{j=1}^{N_x} \mathbf{x}_j = \mathbf{0} \tag{1}$$

where N_y and N_x are the numbers of points in $\mathbf{Y}$ and $\mathbf{X}$ respectively. We denote a match between point $\mathbf{y}_i$ and point $\mathbf{x}_j$ as $\mathbf{y}_i \leftrightarrow \mathbf{x}_j$. In our scheme we assume that every point in $\mathbf{Y}$ is matched to a unique point in $\mathbf{X}$ and that the relationship between each pair of matched points is that of a linear transformation plus noise, encapsulated in the following expression:

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_j + \mathbf{t} + \boldsymbol{\epsilon}_{i,j} \tag{2}$$

The translation component of the linear transformation is modelled by $\mathbf{t}$; other components are captured in $\mathbf{W}$. These are considered to be random variables that define a global transformation which applies to all points in $\mathbf{X}$; local deviations from this are accounted for in the noise term, $\epsilon_{i,j}$, which is assumed to be Gaussian distributed with zero mean and precision $\lambda_{i,j}$. Thus we have a set of random variables, $\Omega = \{\mathbf{W}, \mathbf{t}, \boldsymbol{\lambda}\}$, for which, in this Bayesian model, we must estimate posterior probability distributions, and two sets of observations, $\mathbf{Y}$ and $\mathbf{X}$.

The combination of these posterior distributions and (2) allow us to calculate the probabilities that a particular point in $\mathbf{Y}$ is matched to each of the points in $\mathbf{X}$ (and *vice versa*), so although we assume that each point has a match, all of its match probabilities might be very low, indicating that the truth is that the point has no true match.

Given a proposed set of matches, denoted by $\{\mathbf{y}_i \leftrightarrow \mathbf{x}_j\}$, we define independent match likelihoods over that particular set, giving $\prod_{ij} \mathrm{p}(\mathbf{y}_i \leftrightarrow \mathbf{x}_j \,|\, \Omega)$, where $\prod_{ij}$ denotes the product over the set. Using Bayes' rule the joint posterior distribution of the variables

Algorithm 1 Summary of the new ICP-like algorithm for point pattern matching.

```
make an initial guess at the point mappings
while not converged do
    use the current set of point mappings to make a new, Bayesian estimate of the linear
        transformation
    calculate the matching matrix, $\mathbf{M}$
    select the best set of point mappings from $\mathbf{M}$
end while
```

is given by

$$\mathrm{p}(\Omega \,|\, \{\mathbf{y}_i \leftrightarrow \mathbf{x}_j\}) \propto \mathrm{p}(\Omega) \prod_{ij} \mathrm{p}(\mathbf{y}_i \leftrightarrow \mathbf{x}_j \,|\, \Omega) \tag{3}$$

The prior distributions for the model variables, $\mathrm{p}(\Omega)$, represent any prior knowledge we have about the likely values of these variables and the level of certainty we have in those values. If no knowledge is available to inform these distributions then we might choose flat, or uninformative, priors.

In an exact Bayesian model, calculating the posteriors requires us to integrate the right-hand side of (3) over all possible values of the variables. In this model, as in many others, this integral is intractable, so we use a variational scheme (described in 2.2) to approximate the posteriors.

Having estimated the posteriors we use the posterior expectations of the variables to calculate, for every possible pair of points, the probability that the pair is a match. We construct a $N_y \times N_x$ matching matrix, $\mathbf{M}$, of these probabilities from which we extract the most likely set of matches. Initially this new set of matches is likely to be slightly different from the set used to estimate the posterior distributions. This new proposed set is used as the basis for another round of posterior estimation, which produces another slightly different set of matches, and so on until convergence, resulting in the ICP-like algorithm shown in algorithm 1.

While any $\mathbf{W}$ represents a linear transformation, we wish to control individual elements of it. We achieve this by decomposing $\mathbf{W}$ into a number of separate components, which are treated as independent random variables, and limit the constituent values by defining (possibly problem-dependent) prior distributions over them. For example, in the real example described in section 4.2 we choose to limit the rotation to be very close to zero.

For clarity and because the estimation of the rotation and shear parameters are very similar, we will describe the transformation matrix, $\mathbf{W}$, only in terms of a rotation, $\mathbf{R}$, and a scaling, $\mathbf{D}$:

$$\mathbf{W} = \mathbf{R}\mathbf{D} \tag{4}$$

The scaling matrix, $\mathbf{D}$, is defined as the diagonal matrix

$$\mathbf{D} = \mathrm{diag}(\mathbf{d}) \tag{5}$$

There are a number of different ways to parameterise rotation matrices (e.g. [33]). For mathematical convenience in the Bayesian approximation scheme described in section 2.2 we choose to use the exponential map method; we construct $\mathbf{R}$ from a skew-symmetric matrix, $\mathbf{S}$, as follows:

$$\mathbf{S} = \begin{pmatrix} 0 & s_1 & s_2 \\ -s_1 & 0 & s_3 \\ -s_2 & -s_3 & 0 \end{pmatrix} \tag{6}$$

$$\mathbf{R} = \mathrm{expm}(\mathbf{S}) \tag{7}$$

The matrix exponential function, $\mathrm{expm}(\cdot)$, may be expressed as an infinite series, which we approximate using just the first three terms:

$$\begin{aligned} \mathrm{expm}(\mathbf{S}) &= \mathbf{I} + \mathbf{S} + \frac{\mathbf{S}^2}{2!} + \cdots + \frac{\mathbf{S}^n}{n!} + \ldots \\ &\approx \mathbf{I} + \mathbf{S} + \frac{1}{2}\mathbf{S}^2 \end{aligned} \tag{8}$$

Using this exponential map method the rotation matrix $\mathbf{R}$ represents a rotation through angle $||\mathbf{s}||$ about the vector $\mathbf{s}$ [34, section 13.2.1].

The full set of variables for this model is therefore: $\mathbf{s} = (s_1, s_2, s_3)^{\mathrm{T}}$ (rotation), $\mathbf{d}$ (scaling), $\mathbf{t}$ (translation) and the $\lambda_{i,j}$ (match noise precisions). We now define the prior probability distributions for each of them.

2.1. Priors

As previously described, we assign zero-mean Gaussian priors to each of the noise variables, $\boldsymbol{\epsilon}_{i,j}$, with independent precisions, $\lambda_{i,j}$. With (2) this gives rise to the Gaussian match likelihood

$$\mathrm{p}(\mathbf{y}_i \leftrightarrow \mathbf{x}_j \,|\, \mathbf{W}, \mathbf{t}, \lambda_{i,j}) = \mathcal{N}(\mathbf{y}_i \,|\, \mathbf{W}\mathbf{x}_j + \mathbf{t}, \lambda_{i,j}^{-1}\mathbf{I}) \tag{9}$$

where $\mathcal{N}(\cdot)$ denotes a Gaussian distribution and $\mathbf{I}$ the identity matrix. Each noise precision, $\lambda_{i,j}$, is itself a variable to be estimated and is assigned a conjugate Gamma prior:

$$\mathrm{p}(\lambda_{i,j}) = \mathcal{G}(\lambda_{i,j} \,|\, a_\lambda, b_\lambda) \tag{10}$$

Integrating out the $\lambda_{i,j}$ from the match likelihood results in a Student-t distribution with degrees of freedom $\nu = 2a_\lambda$ and precision a_λ/b_λ. As ν tends to infinity the distribution tends to a Gaussian; as it becomes smaller the distribution becomes heavier tailed until, at 2, the variance becomes infinite.

The translation vector, $\mathbf{t}$, is assigned a Gaussian prior:

$$\mathrm{p}(\mathbf{t}) = \mathcal{N}(\mathbf{t} \,|\, \mathbf{m}_\mathbf{t}, \mathbf{V}_\mathbf{t}) \tag{11}$$

and independent Gaussian priors are assigned to the rotation and scaling variables, $\mathbf{s}$ and $\mathbf{d}$:

$$\mathrm{p}(\mathbf{s}) = \mathcal{N}(\mathbf{s} \,|\, \mathbf{m}_s, \mathrm{diag}(\mathbf{v}_s)) \tag{12}$$

$$\mathrm{p}(\mathbf{d}) = \mathcal{N}(\mathbf{d} \,|\, \mathbf{m}_d, \mathrm{diag}(\mathbf{v}_d)) \tag{13}$$

Prior mean values of $\mathbf{m}_s = \mathbf{0}$ and $\mathbf{m}_d = \mathbf{1}$ denote no rotation and no scaling respectively.

The overall prior is factorised as follows:

$$\mathrm{p}(\Omega) = \mathrm{p}(\mathbf{s})\,\mathrm{p}(\mathbf{d})\,\mathrm{p}(\mathbf{t}) \prod_i \prod_j \mathrm{p}(\lambda_{i,j}) \tag{14}$$

Figure 1 shows a graphical representation of the variables and statistical dependencies of this model. With these priors we now estimate the approximate posterior distributions using the variational Bayesian method.

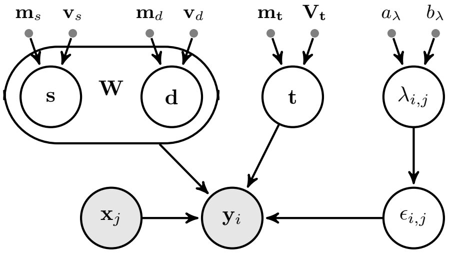


Figure 1 A graphical representation of the model priors. Nodes shaded in grey denote observations; those not shaded denote variables whose posterior distributions must be learned. Small grey points represent parameters of the prior distributions.

2.2. Posteriors

Assume for now that we have a (hypothesised) complete set of matches, i.e. for every $\mathbf{y}_i \in \mathbf{Y}$ there is defined a single match $\mathbf{y}_i \leftrightarrow \mathbf{x}_j$, where $\mathbf{x}_j \in \mathbf{X}$. Based only on these matches and not on the exhaustive set of possible matches, we use the variational Bayesian method to estimate posterior distributions for each of the variables.

Very briefly, variational Bayes (for tutorials see [10, 35] and [36, chapter 10]) seeks approximate posterior distributions $\mathrm{q}(\Omega_i) \approx \mathrm{p}(\Omega_i \,|\, \mathcal{D})$ (where $\Omega_i \in \Omega$ represents one, or one group, of the model's variables and $\mathcal{D}$ the data) that minimise the Kullback-Leibler (KL) divergence [12] between q and p, where

$$\mathrm{KL}(\mathrm{q} \parallel \mathrm{p}) = \int \mathrm{q}(\Omega) \log\left(\frac{\mathrm{q}(\Omega)}{\mathrm{p}(\Omega \,|\, \mathcal{D})}\right) d\Omega \tag{15}$$

The KL divergence is non-negative, and only zero when q and p are equal. An elegant method provided by Waterhouse et al. [37] (see also [11, 38, 39]) exploits the assumed factorisation of the approximate posterior as

$$\mathrm{q}(\Omega) = \prod_i \mathrm{q}(\Omega_i \,|\, \mathcal{D}) \tag{16}$$

and leads to

$$\log(\mathrm{q}(\Omega_i)) = \mathbb{E}_{/\Omega_i}\left[\log(\mathrm{p}(\mathcal{D}, \Omega))\right] \tag{17}$$

where $\mathbb{E}_{/\Omega_i}[\cdot]$ denotes the expectation based on all variables except Ω_i. When conjugate priors are chosen for the variables Ω_i the resulting posterior distributions are of the same family and their parameters are expressed in terms of the expectations of the other variables in the problem. Suitable posterior parameter values are found by iteratively calculating each of the $\mathrm{q}(\Omega_i)$ in terms of the others until convergence. Ghahramani and Beal [40] show that this method converges to a local minimum of $\mathrm{KL}(\mathrm{q} \parallel \mathrm{p})$.

We assume that the approximate posteriors are independent of one another conditional on the data, and are factorised based on the factorisation of the priors shown in (14):

$$\mathrm{q}(\mathbf{s})\,\mathrm{q}(\mathbf{d})\,\mathrm{q}(\mathbf{t})\prod_i\prod_j \mathrm{q}(\lambda_{i,j}) \tag{18}$$

We illustrate the variational method described above by estimating the approximate posterior distribution for the first of the rotation variables, s_1. From (17) we get

$$\log(\mathrm{q}(s_1)) = \mathbb{E}_{/s_1}\left[\log\left\{\left[\prod_{ij}\mathrm{p}(\mathbf{y}_i \leftrightarrow \mathbf{x}_j \,|\, \mathbf{s},\mathbf{d},\mathbf{t},\lambda_{i,j})\right]\mathrm{q}(\mathbf{s})\,\mathrm{q}(\mathbf{d})\,\mathrm{q}(\mathbf{t})\prod_i\prod_j \mathrm{q}(\lambda_{i,j})\right\}\right] \tag{19}$$

Under this expectation all terms not directly dependent on s_1 are constant, so we may rewrite (19) as

$$\log(\mathrm{q}(s_1)) = \mathbb{E}_{/s_1}\left[\log(\mathcal{N}(s_1 \,|\, m_{s_1}, v_{s_1})) + \sum_{ij}\log(\mathcal{N}(\mathbf{y}_i \,|\, \mathbf{W}\mathbf{x}_j + \mathbf{t}, \lambda_{i,j}^{-1}\mathbf{I}))\right] + \mathit{const} \tag{20}$$

Expanding $\mathbf{W}$ using (4,7-8) and absorbing some terms not dependent on s_1 into the constant term results in

$$\begin{aligned}\log(\mathrm{q}(s_1)) = -\frac{1}{2}\mathbb{E}_{/s_1}\Bigg[& s_1^2 v_{s_1}^{-1} - 2s_1 v_{s_1}^{-1} m_{s_1}\\ &+ \sum_{ij}\lambda_{i,j}(\mathbf{t}-\mathbf{y}_i)^{\mathrm{T}}(2\mathbf{S}+\mathbf{S}^2)\mathbf{D}\mathbf{x}_j\Bigg] + \mathit{const}\end{aligned} \tag{21}$$

Expanding $\mathbf{S}$ using (6) and moving these additional terms into the constant gives

$$\begin{aligned}\log(\mathrm{q}(s_1)) = -\frac{1}{2}\Bigg[& s_1^2\Big(v_{s_1}^{-1} + \sum_{ij}\langle\lambda_{i,j}\rangle(\mathbf{y}_i - \langle\mathbf{t}\rangle)^{\mathrm{T}}\mathbf{G}_1\langle\mathbf{D}\rangle\mathbf{x}_j\Big)\\ &- 2s_1\Big(v_{s_1}^{-1}m_{s_1} - \sum_{ij}\frac{\langle\lambda_{i,j}\rangle}{2}(\mathbf{y}_i - \langle\mathbf{t}\rangle)^{\mathrm{T}}\mathbf{H}_1\langle\mathbf{D}\rangle\mathbf{x}_j\Big)\Bigg] + \mathit{const}\end{aligned} \tag{22}$$

where

$$\mathbf{G}_1 = \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{pmatrix} \qquad \mathbf{H}_1 = \begin{pmatrix}0 & -2 & -\langle s_3\rangle\\ 2 & 0 & \langle s_2\rangle\\ -\langle s_3\rangle & \langle s_2\rangle & 0\end{pmatrix} \tag{23}$$

and $\langle a \rangle$ denotes the posterior expectation of a. This is quadratic in s_1, so the approximate posterior for s_1 is the Gaussian distribution shown in (28). Standard results give $\langle s_1^2 \rangle = \Sigma_{s_1} + \langle s_1 \rangle^2$. With

$$\mathbf{G}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{H}_2 = \begin{pmatrix} 0 & \langle s_3 \rangle & -2 \\ \langle s_3 \rangle & 0 & \langle s_1 \rangle \\ 2 & \langle s_1 \rangle & 0 \end{pmatrix} \tag{24}$$

$$\mathbf{G}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{H}_3 = \begin{pmatrix} 0 & \langle s_2 \rangle & -\langle s_1 \rangle \\ \langle s_2 \rangle & 0 & -2 \\ -\langle s_1 \rangle & 2 & 0 \end{pmatrix} \tag{25}$$

similar results are obtained for s_2 and s_3.

Using the same procedure for the remaining variables results in the approximate posteriors shown in the remainder of figure 2. Since the $\mathbf{s}$ and $\mathbf{d}$ variables are all assumed to be posteriorly independent, the posterior expectations $\langle \mathbf{W} \rangle$ and $\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \rangle$ are easily constructed from (4, 6–8):

$$\langle \mathbf{W} \rangle = \left(\mathbf{I} + \langle \mathbf{S} \rangle + \langle \mathbf{S}^2 \rangle \right) \langle \mathbf{D} \rangle \tag{26}$$

$$\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \rangle = \langle \mathbf{D}^2 \rangle \tag{27}$$

since $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}$ and $\mathbf{D}$ is a diagonal matrix.

These expressions do not form a closed solution as each posterior is dependent on the posterior expectations of other variables. Instead they are evaluated iteratively until convergence. In variational approximation procedures the order in which the posteriors are reestimated is important as it can affect the quality of the final solution. We start with the variables closest to the observations in the graphical model (figure 1) and work outwards. Hence we visit the variables in the order $\mathbf{s}$, $\mathbf{d}$, $\mathbf{t}$ and finally the $\lambda_{i,j}$.

Note that the expressions for the variances for $\mathbf{s}$ and the match noise precisions in the $\lambda_{i,j}$ do not preclude negative values. Where this situation arises the posterior distribution is set to the prior for that variable. This only occurs during the initial iterations, not at convergence.

$$\mathrm{q}(s_k) = \mathcal{N}(s_k \,|\, \mu_{s_k}, \Sigma_{s_k}) \tag{28}$$

$$\Sigma_{s_k} = \left[v_{s_k}^{-1} + \sum_{ij} \langle \lambda_{i,j} \rangle (\mathbf{y}_i - \langle \mathbf{t} \rangle)^{\mathrm{T}} \mathbf{G}_k \langle \mathbf{D} \rangle \mathbf{x}_j \right]^{-1}$$

$$\mu_{s_k} = \Sigma_{s_k} \left[v_{s_k}^{-1} m_{s_k} - \frac{1}{2} \sum_{ij} \langle \lambda_{i,j} \rangle (\mathbf{y}_i - \langle \mathbf{t} \rangle)^{\mathrm{T}} \mathbf{H}_k \langle \mathbf{D} \rangle \mathbf{x}_j \right]$$

$$\mathrm{q}(d_q) = \mathcal{N}(d_q \,|\, \mu_{d_q}, \Sigma_{d_q}) \tag{29}$$

$$\Sigma_{d_q} = \left(v_{d_q}^{-1} + \sum_{ij} \langle \lambda_{i,j} \rangle x_{q,j}^2 \right)^{-1}$$

$$\mu_{d_q} = \Sigma_{d_q} \left(v_{d_q}^{-1} m_{d_q} + \sum_{ij} \langle \lambda_{i,j} \rangle \langle a_q \rangle x_{q,j} \right)$$

$$\langle \mathbf{a} \rangle = \left[\mathbf{I} + \frac{1}{2} \begin{pmatrix} -\langle s_1^2 \rangle - \langle s_2^2 \rangle & -\langle s_2 \rangle \langle s_3 \rangle + 2\langle s_1 \rangle & \langle s_1 \rangle \langle s_3 \rangle + 2\langle s_2 \rangle \\ -\langle s_2 \rangle \langle s_3 \rangle - 2\langle s_1 \rangle & -\langle s_1^2 \rangle - \langle s_3^2 \rangle & -\langle s_1 \rangle \langle s_2 \rangle + 2\langle s_3 \rangle \\ \langle s_1 \rangle \langle s_3 \rangle - 2\langle s_2 \rangle & -\langle s_1 \rangle \langle s_2 \rangle - 2\langle s_3 \rangle & -\langle s_2^2 \rangle - \langle s_3^2 \rangle \end{pmatrix} \right]^{\mathrm{T}} (\mathbf{y}_i - \langle \mathbf{t} \rangle)$$

$$\mathrm{q}(\mathbf{t}) = \mathcal{N}(\mathbf{t} \,|\, \mu_t, \Sigma_t) \tag{30}$$

$$\boldsymbol{\Sigma}_t = \left(\mathbf{V}_{\mathbf{t}}^{-1} + \sum_{ij} \langle \lambda_{i,j} \rangle \right)^{-1}$$

$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t \left(\mathbf{V}_{\mathbf{t}}^{-1} \mathbf{m}_{\mathbf{t}} + \sum_{ij} \langle \lambda_{ij} \rangle (\mathbf{y}_i - \langle \mathbf{W} \rangle \mathbf{x}_j) \right)$$

$$\mathrm{q}(\lambda_{i,j}) = \mathcal{G}(\lambda_{i,j} \,|\, \alpha_{\lambda_{i,j}}, \beta_{\lambda_{i,j}}) \tag{31}$$

$$\alpha_{\lambda_{i,j}} = a_\lambda + \frac{D}{2}$$

$$\beta_{\lambda_{i,j}} = b_\lambda + \frac{1}{2} \left(\mathbf{y}_i^{\mathrm{T}} \mathbf{y}_i + \mathbf{x}_j^{\mathrm{T}} \langle \mathbf{W}^{\mathrm{T}} \mathbf{W} \rangle \mathbf{x}_j + \langle \mathbf{t}^{\mathrm{T}} \mathbf{t} \rangle - 2\mathbf{y}_i^{\mathrm{T}} (\langle \mathbf{W} \rangle \mathbf{x}_j + \langle \mathbf{t} \rangle) + 2\langle \mathbf{t} \rangle \langle \mathbf{W} \rangle \mathbf{x}_j \right)$$

Figure 2 Calculations for the approximate posterior distributions for each of the model variables.

2.3. Matching algorithm

The previous section describes how, given a set of point mappings, we use variational Bayesian approximation to estimate the linear transformation. To complete the algorithm we now describe how the current estimate of the transformation is used to determine a new set of point mappings.

We start by constructing a $N_y \times N_x$ matching matrix, $\mathbf{M}$, where element $M_{i,j}$ is the logarithm (for mathematical convenience) of the probability that $\mathbf{y}_i$ is mapped to $\mathbf{x}_j$, based on the posterior expectations of the transformation variables:

$$M_{i,j} = \log(\mathcal{N}(\mathbf{y}_i \,|\, \langle \mathbf{W} \rangle \mathbf{x}_j + \langle \mathbf{t} \rangle, \langle \lambda_{i,j} \rangle^{-1} \mathbf{I})) \tag{32}$$

These values are calculated for every possible match.

For an exact matching scheme we might define Dirichlet priors over each row of $\mathbf{M}$; we know that every $\mathbf{y}_i$ must have a match in $\mathbf{X}$, so $\sum \exp(M_{i,:}) = 1$. We could then contemplate estimating $\mathbf{M}$ variationally. Consider, however, the case of a point $\mathbf{y}_i$ that is truly unmatched. The Dirichlet would identify the most likely match for $\mathbf{y}_i$, but not the absolute probability that the match is true. For a truly unmatched point we would like all its match probabilities (the values in the corresponding row of $\mathbf{M}$ for points in $\mathbf{Y}$ or column of $\mathbf{M}$ for points in $\mathbf{X}$) to be very small. For this reason, and because we require the one-to-one correspondence to be bidirectional, the Dirichlet is not appropriate here and we must resort to cruder means to extract the best set of matches from $\mathbf{M}$.

Ideally we would select from this matrix the set of mappings $\mathbf{y}_i \leftrightarrow \mathbf{x}_j$ that maximises $\sum_{ij} M_{i,j}$, in other words finding a permutation of the columns of $\mathbf{M}$ that maximises the sum of the values on its main diagonal. But this is in itself a hard problem. An obvious method is to select the element of $\mathbf{M}$ with the highest value, $M_{a,b}$, form a mapping between $\mathbf{y}_a$ and $\mathbf{x}_b$, remove row a and column b of $\mathbf{M}$ from consideration and then repeat the process until all points in $\mathbf{Y}$ (the smaller set) are mapped. This is the *greedy algorithm.* Although initially attractive, it is very prone to converging prematurely on local optima. An alternative, the *random permutation algorithm*, is to visit each row of $\mathbf{M}$ in a random order and map the point in $\mathbf{Y}$ corresponding to that row with the point in $\mathbf{X}$ corresponding to the element in that row that has the highest value. This leads

Algorithm 2 Inexact Bayesian point set matching

```
randomly initialise $\mathbf{s}$, $\mathbf{d}$ (hence $\mathbf{W}$) and $\mathbf{t}$
while not converged do
  calculate $\mathbf{M}$ using (32)
  select the set of matches from $\mathbf{M}$ (2.3)
  for the current set of matches do
    update the posteriors for $\mathbf{s}$ using (28)
    update the posteriors for $\mathbf{d}$ using (29)
    calculate $\langle\mathbf{W}\rangle$ and $\langle\mathbf{W}^{\mathrm{T}}\mathbf{W}\rangle$ using (26–27)
    update the posterior for $\mathbf{t}$ using (30)
    for $i = 1$ to $N_y$ do
      for $j = 1$ to $N_x$ do
        update the posterior for $\lambda_{i,j}$ using (31)
      end for
    end for
  end for
end while
calculate $\mathbf{M}$ using (32)
select the final set of matches from $\mathbf{M}$ (2.3)
```

to a wider exploration of the search space and significantly increases the frequency with which the global optimum is found. However, in noisy datasets that contain unmapped points this algorithm tends not to converge on a single, stable solution. We use the *random permutation algorithm* for the first iterations to ensure wider exploration and then switch to the *greedy algorithm* to force a stable convergence.

The overall method alternates between variationally estimating the linear transformation and finding the best set of point matches from $\mathbf{M}$ until the match set converges, as is shown in algorithm 2. It is not guaranteed to find the optimum solution, but we run it several times on a given dataset, with different random starting positions for the transformation variables, and select the solution that results in the highest final value of $\sum_{ij} M_{i,j}$.

2.4. Unmatched points

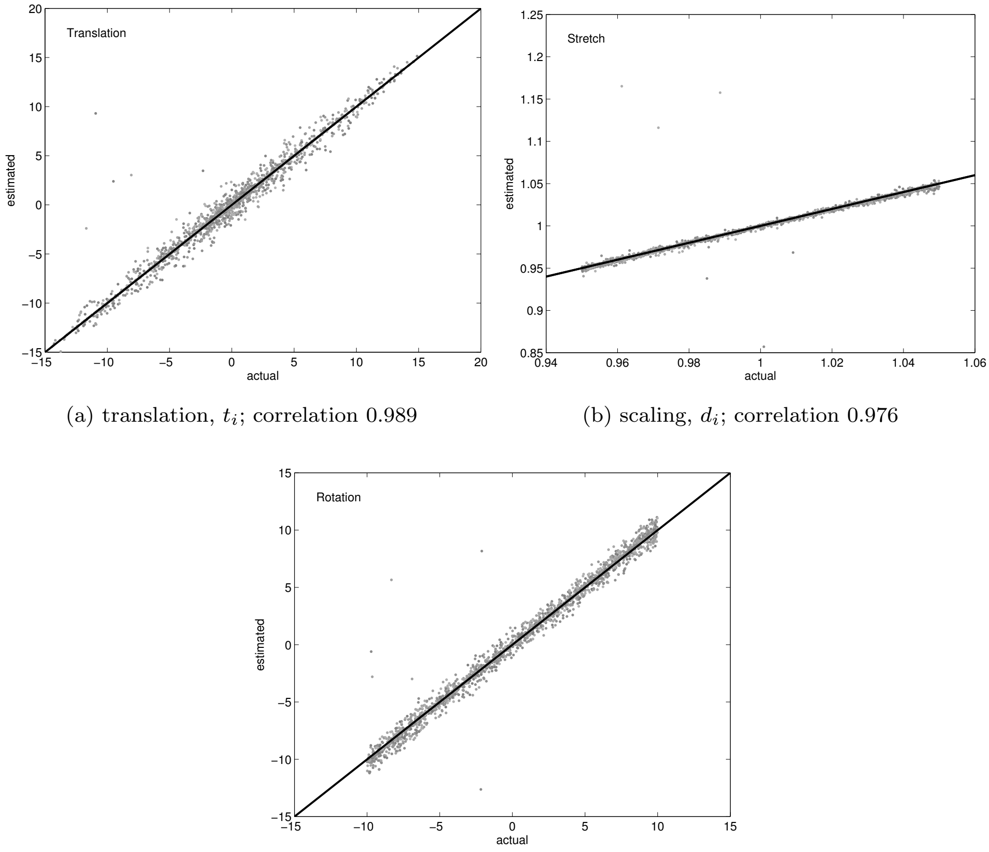
We could use the probabilities in the matching matrix corresponding to the final set of point mappings to estimate which points are truly unmatched, but a more intuitive decision may be made based on the associated $\lambda_{i,j}$ values.

The standard deviation corresponding to these precisions, $\sigma_{i,j} = \sqrt{1/\lambda_{i,j}}$, is a measure of the distance between $\mathbf{y}_i$ and $\mathbf{x}_i$ after the linear transformation has been taken into account. A very small $\sigma_{i,j}$ means that the points are closely aligned and are thus likely to be a true match, while a large value indicates a less likely match. Problem-specific information may indicate a specific threshold to differentiate between matched and unmatched points; this method is used for the real data described in section 4.2. Alternatively, it is often the case that plotting the $\sigma_{i,j}$ values in ascending order highlights an obvious transition from which a threshold may be derived. The latter is the method used for the synthetic data described in the next section.

3. Illustration: synthetic data

In the first instance we demonstrate the effectiveness of the method on pairs of point sets that are related by known transformations, with unmatched points but no noise. First 20,000 points are uniformly randomly sampled across a 1,000 unit cube centred on the origin. The points falling within the central 100 unit cube (approximately 200 points) make up the first set of points. The larger cube is then randomly scaled, rotated and translated with respect to the origin and a second set of points extracted from the 100 unit cube centred on the origin with the same orientation as the first. This simulates the real case described in section 4.2 where the points are extracted from 3-dimensional images of a substance photographed before and after the substance has been stretched and moved/rotated with respect to the camera.

1,000 pairs of point sets were generated using different, randomly-generated transformations, with the rotation angles uniformly sampled from a range of ±10°, the scaling variables in the range 0.95 to 1.05 and translations within a sphere of radius 20 units. The prior parameters were set as follows: $\mathbf{m}_s = \mathbf{m}_t = \mathbf{0}$, $\mathbf{m}_d = \mathbf{1}$, $\mathbf{v}_s = \mathbf{1}$, $\mathbf{V}_t = 40^2\mathbf{I}$,



(a) translation, t_i; correlation 0.989

(b) scaling, d_i; correlation 0.976

(c) rotation angle (degrees); correlation 0.995

Figure 3 The values of the estimated transformation variables plotted against the actuals for the 1,000 tests datasets, with no noise. The diagonal black lines mark the locations of estimate=actual.

$\mathbf{v}_d = 10^{-3}\mathbf{I}$, $a_\lambda = 1$ and $b_\lambda = 1$. The overlap between the two point sets was, on average, 86% of the number of points in $\mathbf{X}$, with a standard deviation of 5.

The model was run for 10 repetitions against each dataset and the best result, i.e. that with the highest $\sum_{ij} M_{i,j}$ value, within the 10 repetitions selected as the final solution in each case. Each run was for 100 iterations and the *greedy algorithm* was used for selecting point mappings for the last 50 iterations.

For each of the 1,000 tests we used a $\lambda_{i,j}$ threshold of 0.05 (found empirically) to deter-

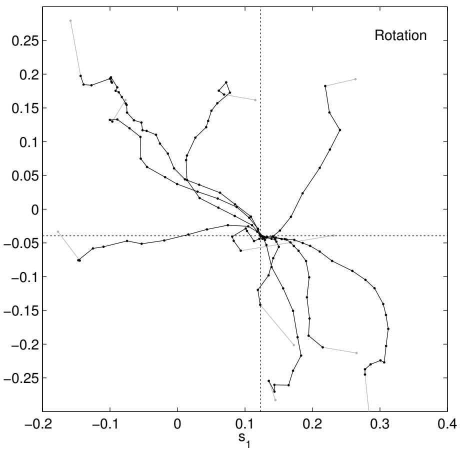


Figure 4 Convergence of the rotation variables s_1 and s_2 over each of the 10 repetitions for one example dataset. The cross-hairs identify the location of the true values and the random starting points are shown in grey.

mine which points were estimated to be matched and which unmatched. The mappings for the matched points were compared with the true mappings; in 946 of the 1,000 tests the mappings were exactly correct, that is all the truly matched points were correctly matched and all the truly unmatched points were correctly identified as being unmatched.

Figure 3 shows the values estimated by the model against the true values for the rotation, scaling and translation variables. The correlation between estimates and actuals is very good for this range of transformations, but it does deteriorate for larger transformations as the search space expands and the overlap between the point sets diminishes. This would be mitigated by increasing the number of repetitions executed for each dataset.

As an example, figure 4 shows, for one of the test datasets, how the rotation variables s_1 and s_2 converge on the same values in each of the 10 repetitions. In this case they converge on the true solution, as identified by the cross-hairs.

Figure 5 shows a histogram of the number of times the best solution is found within the 10 repetitions for each of the 1,000 test datasets. In 165 cases the same result is found in all 10 repetitions and in 856 cases the best result is found in more than 50% of the repetitions.

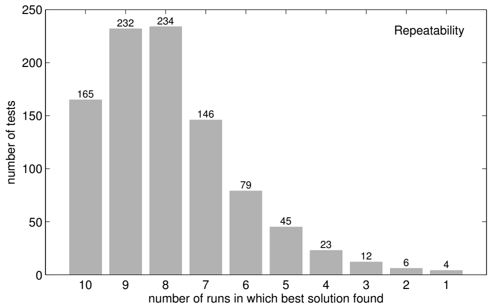


Figure 5 A histogram which shows how often the best result is identified in 10 repetitions across 1,000 test datasets.

Using the first 100 test datasets, the points in $\mathbf{Y}$ were perturbed by varing amounts of noise. The perturbations were generated from the Gaussian distribution $\mathcal{N}(0, \sigma^2)$, where the standard deviation, σ, was set successively to the integer values in the range 0 to 10. The mean distance between 200 points uniformly distributed within a 100^3 unit cube is approximately 64 units, so a standard deviation of 10 is significant. The model was run for 10 repetitions for each dataset and each value of σ, and the best result selected as before. Figure 6 shows how the standard deviations of the transformation variables' posterior distributions increases as the magnitude of the noise increases (results are very similar for the rotation and scaling variables, though with much smaller standard deviations), showing how the model becomes less certain of the estimates where there is more noise. It is often remarked (e.g. [41, 42, 43]) that there is a tendency for variational models to underestimate the uncertainty, so the values should be treated as a lower bound. As the noise increases, so the proportion of correct matches decreases, as is shown in figure 7.

Finally, in order to investigate how the degree of overlap between the two point sets affects the quality of the matching, we generated two identical point sets and progressively replaced points in $\mathbf{Y}$ with new random points. We performed 10 repetitions of matching and recorded the number of times all the points in the overlap region were correctly matched. The results are shown in figure 8 and show that, generally, as the overlap proportion decreases, the proportion of correct matches also decreases, but the decrease does not until the overlap region drops to about 50%.

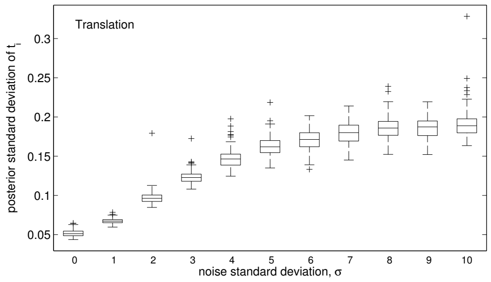


Figure 6 The change in posterior standard deviations for the translation variable, **t**, as the noise increases. Each box indicates the median and 25th and 75th percentiles of the standard deviations across the 100 tests.

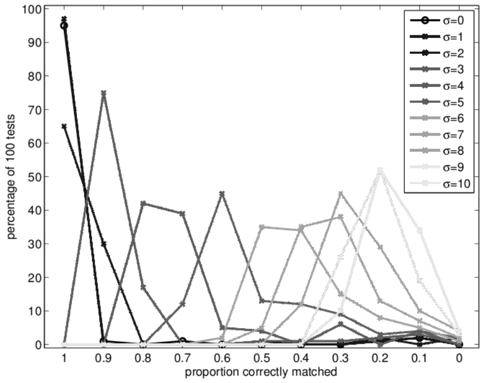


Figure 7 For the 100 tests, the proportion of correct matches for different noise levels. As the standard deviation of the noise, σ, increases, so the proportion of correct matches decreases.

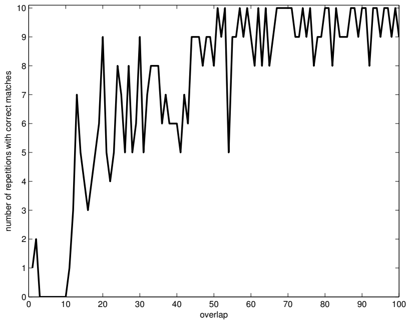


Figure 8 For point sets containing 100 points, this graph shows the effect of reducing the overlap between the two sets. For each size of overlap, matching was performed for 10 repetitions and the number of repetitions in which the overlap region was correctly matched is plotted.

4. Results: real data

The model is demonstrated on two sets of real data: 2-dimensional images from the CMU house dataset [44] (a series of 111 images of a toy house rotating in 3D) and 3-dimensional microscopy images of a sample of cartilage.

4.1. 2D CMU house

The locations of the same thirty features from each of the CMU house images were manually recorded. For each image a further ten locations were selected uniformly randomly across the image as noise. Figure 9 shows the first image in the sequence on the left in both columns and image 20 on the right, with the locations of true features marked as dots and those of the noise features as crosses. The top pair of plots record the matches between images 1 and 20 made using the standard ICP algorithm[1] [1, 7]; the bottom pair record the highest-probability matches (those with $\log_e$ probability greater than -5.5) identified by the algorithm described in this paper. Both algorithms appear to perform similarly, but our algorithm allows selection of well-matched points on the basis of their posterior match probability. Note that the low probability matches are associated with the noise points.

These matching images do not highlight the differences between the two algorithms. To illustrate these differences, the features from image 1 were translated, rotated and (for the new method) scaled according to the transformations estimated by the two algorithms. Figure 10 shows them (as dots) superimposed on image 20 and the features from that image (plotted as squares). Matched features are joined by a line. For the model, each match has an associated probability (one standard deviation is shown as a circle around each transformed feature) and we can see that it has achieved a strong set of matches for those features located on and around the main roof of the house, a generally good set of matches for the rest of the true features and generally low probability matches for the noise features.

[1] We used Per Bergström's Matlab version of the ICP algorithm from `http://www.mathworks.com/matlabcentral/fileexchange/12627-iterative-closest-point-method`

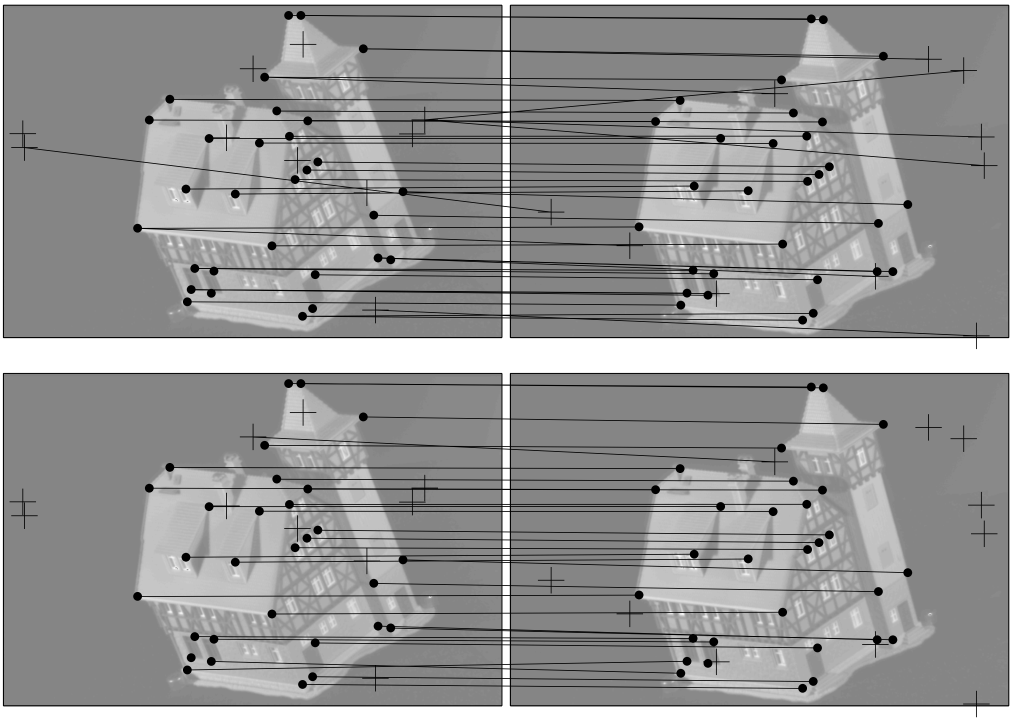
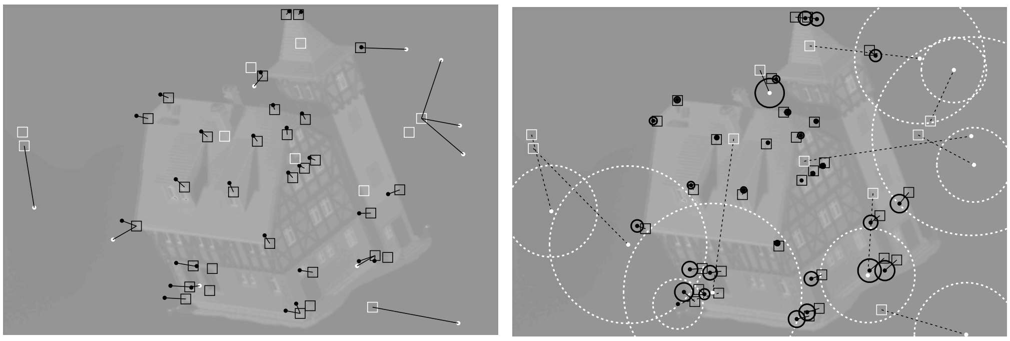


Figure 9 CMU house images 1 (left) and 20 (right) with matches made by the standard ICP algorithm (top) and this new model (bottom). Dots mark the true features; crosses the added noise features.

Using the new method, low probability matches have little influence on the transformation. ICP tries to find a transformation that best aligns all the points, so the noise features have as much influence as the genuine features. The results of this influence are shown more clearly in figure 11 where some of the features associated with image 1 were removed before matching, as if the lower part of the image were occluded. All points below the horizontal dashed line in figure 11b were removed, amounting to 20 of the 30 true features and 4 of the 10 noise features. The probabilistic model has again “locked onto” the features on and around the main roof with high-probability matches. The effect of the occlusion has clearly had a significant effect on the quality of the matches achieved by ICP.



(a) ICP

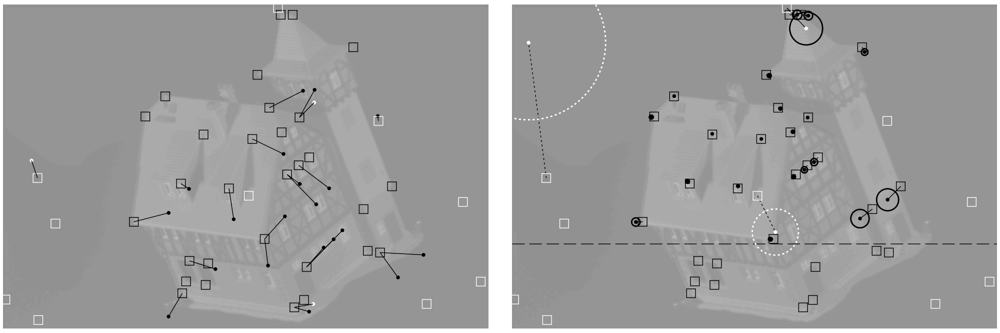
(b) new model

Figure 10 CMU house image number 20 marked with the true features for this image as black squares and the noise features as white squares. The features from image 1 have been transformed according to the results of each of the algorithms and are overlaid as black dots for true features and white for noise. Black lines indicate the matches obtained from each algorithm. Figure (b) is also overlaid with circles indicating one standard deviation of the match probability distribution; low probability matches are shown as dotted lines and circles. Note how the low probability matches are associated with the (white) noise points.

4.2. 3D microscopy images of cartilage cells

This real data comprises pairs of 2 photon fluorescence image stacks of cartilage captured by a multiphoton microscope [45]. The xy plane is approximately parallel to the surface of the cartilage and the individual images in the stack are “slices” obtained by moving the focal plane progressively deeper in the z direction. An example image from one of these stacks is shown in figure 12a; the in-focus cartilage cells are clearly discernable as dark, approximately elliptical shapes against a lighter background. Each cell appears in the same position in a number of adjacent images in the stack and does not, generally, have any uniquely identifiable features.

The locations of the cells are identified by an automated image processing program which fails to identify some cells and spuriously identifies non-existant cells. The two point sets are the locations of the centres of the cells identified in pairs of z-stacks recorded *before* and *after* the tissue has been subjected to a stretch along the x axis. Photo-bleaching leads to *after* stacks that are notable shorter in the z direction than the



(a) ICP

(b) new model

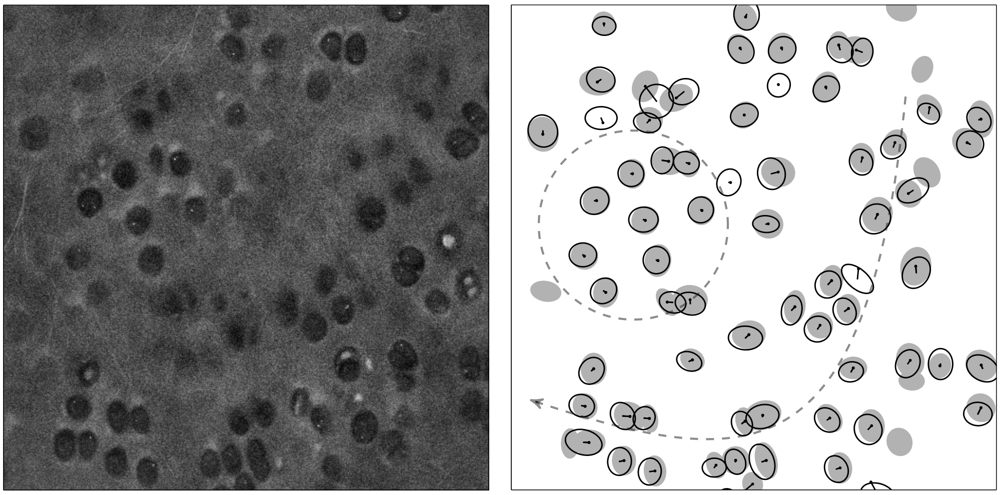
Figure 11 CMU house matches with occlusion.All points from image 1 below the black dashed line shown in (b) were removed prior to matching. The new model has achieved a very good match for the features on and around the main roof section, while ICP has not achieved a reasonable match. See figure 10 for the key to the symbols used.

corresponding *before* stacks. In addition to any stretch applied to the samples, the nature of the experimental procedure means that the sample may also have rotated slightly and moved in relation to the original field of view.

The tissue between the rather sparse cells is inhomogenous, causing cells at different locations to be perturbed in different directions. The aim is to perform matching on the cells so that bio-medical researchers can study these inhomogenous perturbations [45]. Since the registration process also measures the size and orientation of the cells, matching provides additional information regarding the changes to the cells themselves.

Figure 12b shows some results from an experiment in which a sample was strained by approximately 5% along the x axis. The figure shows the matching results associated with the example image shown in figure 12a, with manually-added annotations indicating areas of localised consistency in the cell movements. Each point set contained 800 cells and 10 repetitions of 400 iterations of the algorithm was used.

The priors for most variables were set as for the synthetic data, apart from the translation expectation, $\mathbf{m}_t$, which was set to align the tops of the two stacks (i.e. to align the images at the surface of the cartilage). The estimates made by the model are as follows (in order



(a) original *before* image

(b) cell mappings

Figure 12 Real results for cartilage strained by 5% in tension. An example original *before* image is shown in (a); the matching results for that image are shown in (b). Grey, filled-in ellipses are the cells identified in the *before* image. Black ellipses are those identified in the *after* stack, after the estimated transformation has been reversed. The centre of each matched *before* cell is marked by a dot from which emanates a line linking it to the centre of its mapped *after* cell. The dashed lines are annotations indicating a region of no perturbation (the circle) and a direction (the arrow) in which the cells are perturbed in a consistent manner.

of x, y, z): rotation angles (in degrees) 0.0005, 0.0042, -0.0019; scaling 1.0344, 0.9365, 1.0290; translation -4.3, -24.5, 3.5 pixels (each image is 512 pixels square and there are 81 images in the *before* stack). The translation variables have a posterior standard deviation of 0.17; those for the rotation and scaling are of the order of 10^{-3}. From these results we can see that the stretch detected by the model is about 3.4% in the x direction (rather than the 5% estimated by the experimenter) and there is about a 6.4% compression in the y direction. Having inspected the images, we surmise that the apparent stretch in the z direction is caused by non-linear bowing of the sample at the top and bottom edges.

5. Conclusions

This new model has been described in terms of point locations only. Further attributes such as colour or shape associated with these locations are easily incorporated into the Bayesian framework.

The computation time taken by the algorithm scales quadratically with the number of points in $\mathbf{Y}$ (the smaller of the two point sets), but linearly with the number of points in $\mathbf{X}$. Performance enhancements can be gained by reducing the number of points in $\mathbf{Y}$, perhaps by random sampling to ensure that the subset is unlikely to contain only points that are truly unmatched. As an example, 100 iterations of the (untuned) Matlab implementation of the model running for two points sets containing 500 points each took approximately 130 seconds on a 12 core Linux server. However, we note that the algorithm converged before the 100 iterations were completed and stopping at convergence would reduce the computation time. While standard ICP is much faster (about 14 seconds on the same server), the two algorithms are not directly comparable (for example, ICP does not allow for scaling). We draw attention to both the quality of match and the measure of match quality provided by this model, especially where some points are occluded.

Future work is focussed on extending this methodology to affine and non-linear transformations.

Acknowledgments

The authors would like to thank Dr Jessica Mansfield for assistance with equipment design and multiphoton microscopy.

This work was funded by a grant from the EPSRC's Bridging the Gaps initiative and the collaboration was formed through the Exeter Imaging Network.

References

[1] Y. Chen, G. Medioni, Object modeling by registration of multiple range images, in: Proceedings of the 1991 IEEE Conference on Robotics and Automation, volume 10, Sacramento, California, pp. 145–155.

[2] Y. Hongbo, Object tracking using point matching based on MCMC, in: Proceedings of the 2009 IEEE WRI World Congress on Computer Science and Information Engineering, volume 4, pp. 182–186.

[3] M. Toivanen, J. Lampinen, Incremental object matching and detection with Bayesian methods and particle filters, Computer Vision, IET 5 (2011) 201–210.

[4] E. Serradell, J. Kybic, F. Moreno-Noguer, P. Fua, Robust elastic 2D/3D geometric graph matching, Proceedings of the Society of Photo-Optical Instrumentation Engineers 8314 (2012).

[5] A. Abdulkader, Parallel Algorithms for Labelled Graph Matching, Ph.D. thesis, Colorado School of Mines, 1998.

[6] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, International Journal of Pattern Recognition and Artificial Intelligence 18 (2004) 265–298.

[7] P. Besl, N. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (1992) 239–256.

[8] D. Mackay, Ensemble learning and evidence maximisation, Technical Report, Cavendish Laboratory, University of Cambridge, 1995.

[9] D. Mackay, Ensemble learning for hidden Markov models, Technical Report, Cavendish Laboratory, University of Cambridge, 1997.

[10] M. Jordan, Z. Ghahramani, T. Jaakkola, L. Saul, An introduction to variational methods for graphical models, Machine Learning 37 (1999) 183.

[11] H. Attias, A variational Bayesian framework for graphical models, Advances in Neural Information Processing Systems 12 (2000) 209–215.

[12] S. Kullback, R. Leibler, On information and sufficiency, Annals of Mathematical Statistics 22 (1951) 79–86.

[13] T. Cover, J. Thomas, Elements of Information Theory, John Wiley & Sons, Inc, 1991.

[14] W. Tsai, K. Fu, Error-correcting isomorphisms of attributed relational graphs for pattern analysis, IEEE Transactions on Syst. Man. Cybern. 9 (1979) 757–768.

[15] M. Fischler, R. Elschlager, The representation and matching of pictorial structures, IEEE Transactions on Computing 22 (1973) 67–92.

[16] J. Kittler, E. Hancock, Combining evidence in probabilistic relaxation, International Journal of Pattern Recognition and Artificial Intelligence 3 (1989) 29–51.

[17] H. Almohamad, S. Duffuaa, A linear programming approach for the weighted graph matching problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 522–525.

[18] A. Rangarajan, E. Mjolsness, A Lagrangian relaxation network for graph matching, IEEE Transactions on Neural Networks 7 (1996) 1365–1381.

[19] S. Gold, A. Rangarajan, A graduated assignment algorithm for graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 377–388.

[20] B. Luo, E. Hancock, Structural graph matching using the EM algorithm and singular value decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 1120–1136.

[21] S. Granger, X. Pennec, Multi-scale EM-ICP: a fast and robust approach for surface registration, in: Proceedings of the 2002 European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp. 418–432.

[22] B. Jian, B. Vemuri, A robust algorithm for point set registration using mixture of Gaussians, in: Proceedings of the 2005 IEEE International Conference on Computer Vision (ICCV), volume 2, pp. 1246–1251.

[23] B. Jian, B. Vemuri, Robust point set registration using Gaussian mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1633–1645.

[24] A. Myronenko, X. Song, Point set registration: coherent point drift, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 2262–2275.

[25] B. Xiao, X. Gao, D. Tao, X. Li, HMM-based graph edit distance for image indexing, International Journal of Imaging Systems and Technology 18 (2008) 209–218.

[26] J. Zhu, S. Du, Z. Yuan, Y. Liu, L. Ma, Robust affine iterative closest point algorithm with bidirectional distance, IET Computer Vision 6 (2012) 252–261.

[27] S. Du, N. Zheng, S. Ying, J. Liu, Affine iterative closest point algorithm for point set registration, Pattern Recognition Letters 31 (2010) 291–799.

[28] J. Naylor, A. Smith, Applications of a method for the efficient computation of posterior distributions, Journal of the Royal Statistical Society. Series C 31 (1982) 214–225.

[29] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equations of state calculations by fast computing machines, Journal of Chemical Physics 21 (1953) 1087–1092.

[30] W. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1970) 97–109.

[31] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 721–741.

[32] N. Gordon, D. Salmond, A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, in: IEE Proceedings-F, volume 140.

[33] F. Grassia, Practical parameterization of rotations using the exponential map, Journal of graphics tools 3 (1998) 29–48.

[34] K. Mardia, P. Jupp, Directional statistics, John Wiley & Sons, Ltd, 2000.

[35] H. Lappalainen, J. Miskin, Advances in Independent Component Analysis, Springer-Verlag, Berlin, pp. 75–92.

[36] C. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2006.

[37] S. Waterhouse, D. MacKay, A. Robinson, Advances in Neural Information Processing Systems 7, MIT Press, pp. 351–357.

[38] M. Beal, Z. Ghahramani, The variational Bayesian EM algorithm for incomplete data: with appli-

cation to scoring graphical model structures, in: Bayesian Statistics, volume 7, Oxford University Press, 2002.

[39] M. Beal, Variational algorithms for approximate Bayesian inference, Ph.D. thesis, University College London, 2003.

[40] Z. Ghahramani, M. Beal, Advances in Neural Information Processing Systems, MIT Press, pp. 507–513.

[41] R. Turner, M. Sahani, Bayesian time series models, Cambridge University Press, pp. 104–124.

[42] D. Mackay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

[43] J. Christmas, R. Everson, Robust autoregression: Student-t innovations using variational Bayes, IEEE Transactions on Signal Processing 59 (2011) 48–57.

[44] Carnegie Mellon University, `http://vasc.ri.cmu.edu/idb/html/motion/house/`, publication date unknown. Last accessed 3rd February 2014.

[45] J. Bell, J. Christmas, J. Mansfield, R. Everson, C. Winlove, Micromechanical response of articular cartilage to tensile load measured using nonlinear microscopy, Biomaterials (accepted, 3 Feb 2014) (2014).