2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

# An Integrated Probabilistic Model for Scan-Matching, Moving Object Detection and Motion Estimation

Joop van de Ven, Fabio Ramos and Gian Diego Tipaldi

*Abstract*— **This paper presents a novel framework for integrating fundamental tasks in robotic navigation through a statistical inference procedure. A probabilistic model that jointly reasons about scan-matching, moving object detection and their motion estimation is developed. Scan-matching and moving object detection are two important problems for full autonomy of robotic systems in complex dynamic environments. Popular techniques for solving these problems usually address each task in turn disregarding important dependencies. The model developed here jointly reasons about these tasks by performing inference in a probabilistic graphical model. It allows different but related problems to be expressed in a single framework. The experiments demonstrate that jointly reasoning results in better estimates for both tasks compared to solving the tasks individually.**

## I. INTRODUCTION

Deployment of robotic systems in complex environments, such as the DARPA Grand Challenge, requires robust techniques for localisation and mapping. Accurate position estimation is one of the primary requirements for achieving robust operation. Additionally, urban environments are characterised by the existence of dynamic objects such as vehicles or people, whose detection and motion estimation is crucial for safety. These localisation, detection and tracking problems can be formulated as two well-known tasks in robotics: 1) data association for estimating the robot movement from ranging sensors and 2) detection and motion estimation of moving objects. Both problems have been studied in great detail in mobile robotics, as they are central to a number of important problems. To date, however, these tasks have generally been considered separately.

This paper describes an approach that performs both tasks simultaneously, as a joint statistical inference process. It builds on graphical model techniques and specifically CRF-Matching [19] and CRF-Clustering [23]. The advantage of formulating the problem as a joint inference process is that it allows the transmission of information, from the model performing scan-matching, to the model performing motion detection and vice-versa.

There are two main contributions in this paper. First, it provides a new formulation for the problems of scan-matching, moving object detection and motion estimation as a joint statistical inference procedure. Second, it presents an algorithm to perform Maximum *a posteriori* (MAP) inference in a graphical model with both discrete and continuous random variables.

This paper is organised as follows: Section II presents an overview of existing solutions to the association and moving object detection problems. Section III discusses the basics of Conditional Random Fields (CRF). This is followed by the definition of the joint graphical model in Section IV. Section V then compares the results of the model with current state of the art scan-matching and moving object detection implementations most notably, CRF-Matching and CRF-Clustering. Finally, conclusions and directions for future work are presented in Section VI.

## II. RELATED WORK

Matching of laser range scans is generally performed using the Iterative Closest Point (ICP) algorithm or one of its many variants [14]. The algorithm iteratively minimises the distance between laser points in one scan and corresponding laser points in a consecutive scan by rotating and translating one of the scans. Usually a nearest neighbour approach is used to associate laser points. Despite having good computational performance, ICP has several problems. Firstly it is sensitive to the initial estimated scan offset. Secondly it does not provide a probabilistic interpretation for rotation and translation or even for point correspondences. Finally, it does not take into account other properties of the scans such as shape, or intensity. Various variants of the ICP algorithm have dealt with one or more of the above limitations (see for example [8] or [5]). However, there is no variant of ICP that address all these issues.

To overcome these limitations, a probabilistic data association model was proposed in [19]. The model is based on Conditional Random Fields (CRF) where features representing different properties of the scans can be incorporated and learned from data. Experimental results demonstrated the benefits of the approach for large transformations, with no prior initialisation. The current paper therefore extends the CRF-Matching approach by incorporating motion detection for each laser point and allowing for direct computation of the translation and rotation of objects (sets of laser points).

The detection and tracking of moving objects (DATMO) problem has been extensively studied [2], in different scenarios, and using different sensors. Related work is discussed with an emphasis on detecting moving objects from a moving platform using a laser range finder. The related work is divided into two categories; algorithms that perform detection and algorithms that perform segmentation and tracking.

J. van de Ven and F. Ramos are with the ARC Centre of Excellence for Autonomous Systems, Australian Centre for Field Robotics, University of Sydney 2006, NSW, Australia {j.vandeven, f.ramos}@acfr.usyd.edu.au

G. D. Tipaldi is with the Social Robotics Lab, Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany tipaldi@informatik.uni-freiburg.de

The detection algorithms address the problem in terms of separating the data into static and dynamic clusters. Dynamic points are used for tracking of dynamic objects while the static points are used to obtain better motion estimates for the moving platform.

Hähnel *et al.* [9] detect moving points in range data using an Expectation-Maximisation (EM) based approach. The algorithm maximises the likelihood of the data using a hidden variable expressing the nature of the points (static or dynamic).

In [26], two separate maps are maintained; one for the static part of the environment and one for the dynamic. The maps employ a modified occupancy grid framework which also infers if points are static or dynamic.

Rodriguez-Losada and Minguez [20] improve data association for the ICP algorithm. They introduce a new metric which models dynamic objects to better reflect the real motion of the robot. Their approach however, does not distinguish moving objects from outliers.

The second category of algorithms focuses on object segmentation and tracking. Anguelov *et al.* [1], [4] detect moving points using simple differencing and then apply a modified EM algorithm for clustering the different objects.

In [25] an integrated solution to the mapping and tracking problem is defined: static points are used for mapping and dynamic points for tracking. Data differencing and consistency-based motion detection [24] is used for the detection and segmentation of dynamic points. Points are classified as static or dynamic and clustered into segments; when a segment contains enough dynamic points is considered dynamic. Montesano *et al.* [16] subsequently improve the classification procedure of [25] by jointly solving it in a sequential manner.

Schulz *et al.* [22] use a feature based approach to detect moving objects; the features used are the local minima of the laser data. The objects are then tracked using a joint probabilistic data association filter (JPDAF).

The focus for most of these approaches is on tracking different objects under different hypothesis. The detection part, however, is mainly based on heuristics such as scan differencing. The detection routines observe the actual position of the object and the velocities are computed by the tracking algorithm. CRF-Clustering [23] on the other hand, applies a feature based graphical model to the data. This allows it to reason about the underlying motion of points; it detects and tracks the objects in a single framework, iteratively improving the estimates.

The work presented in [10] allows different, state-of-the-art algorithms, to be combined in a cascaded/sequential framework. The approach treats each algorithm as a "black-box" though it allows limited unidirectional interaction through feature vectors. The model proposed in this paper on the other hand, integrates and extends the CRF-Clustering and CRF-Matching approaches into a single framework. Motion estimates are computed in-line with motion detection and data association, allowing for joint reasoning enhanced estimates.

## III. PRELIMINARIES

### A. Conditional Random Fields

Conditional Random Fields [12] (CRFs) are undirected graphical models that represent probability distributions; the vertices of the graph index random variables while the edges of the graph capture relationships between variables. In the case of CRFs the distribution is conditional (a *probabilistic discriminative model*); the hidden variables $\mathbf{x} = \langle x_1, x_2, ..., x_n \rangle$ are *globally* conditioned on the observations $\mathbf{z}$; $p(\mathbf{x} \mid \mathbf{z})$.

Let $\mathscr{C}$ be the set of all cliques of the graph. The distribution must then factorise as a product of clique potentials $\phi_c(x_c, \mathbf{z})$, where $c \in \mathscr{C}$ and $x_c$ are the hidden variables of the clique. Clique potentials are commonly expressed by a log-linear combination of feature functions, $\phi_c(x_c, \mathbf{z}) = \exp\left(\mathbf{w}_c^T \cdot \mathbf{f}_c(x_c, \mathbf{z})\right)$, resulting in the definition of a CRF as:

$$p(\mathbf{x} \mid \mathbf{z}) = \frac{1}{Z(\mathbf{z})} \exp\left(\sum_{c \in \mathscr{C}} \mathbf{w}_c^T \cdot \mathbf{f}_c(x_c, \mathbf{z})\right). \quad (1)$$

Here $Z(\mathbf{z}) = \sum_x \exp(\sum_{c \in \mathscr{C}} \mathbf{w}_c^T \cdot \mathbf{f}_c(x_c, \mathbf{z}))$ is the partition function; it normalises the exponential ensuring Equation 1 is a proper distribution. $\mathscr{C}$ is again the set of all cliques in the graph. $\mathbf{w}_c$ are parameters (or weights). The feature functions $\mathbf{f}_c$ extract feature vectors given the value of the clique variables $x_c$ and observations $\mathbf{z}$.

### B. Parameter Learning

The weights $\mathbf{w}_c$ express the relative importance of each feature function. As such they play an important role in determining the shape of the distribution. The weights are learnt by maximising the conditional likelihood (Equation 1) given labelled training data. In our case, this is computationally intractable as the partition function $Z(\mathbf{z})$ sums over the entire state space of all hidden variables. A typical graph in the experiments contains on average 600 hidden variables with an approximate total number of states of 100000.

We therefore use maximum pseudo-likelihood learning [3]. Maximum pseudo-likelihood learning approximates the joint by considering, for each hidden variable, only its neighbours in the graph: the Markov blanket. As a result, the partition function is approximated by summing over the state space of individual hidden variables.

### C. Inference

The model defined in the next section is a joint distribution over data association, motion detection and motion clustering hidden variables. The desired outcome of inference is a configuration of the hidden variables for which the joint distribution achieves its maximum; a maximum *a posteriori* (MAP) inference problem. The proposed algorithm is a variation on Max-Product Loopy Belief Propagation.

Belief Propagation (BP) [18] is a class of inference algorithms in which each node sends messages to each of its neighbours in the graph. The messages convey what a node *believes* its neighbours' state should be given its own state. The received messages together with a node's own belief are then used to compute the MAP configuration.

Construction of the messages is performed using the Max-Product algorithm [18] as follows:

$$m_{ij}(x_i) = \max_{x_j} \left( \phi(x_j, \mathbf{z}) \phi(x_i, x_j, \mathbf{z}) \prod_{k \in \mathcal{N}(j) \setminus i} m_{kj}(x_j) \right), \quad (2)$$

where $m_{ij}(x_i)$ is the message node $j$ sends to node $i$. $\phi(x_j, \mathbf{z})$ is the local clique potential for node $j$. Local clique potentials represent what a node believes its state is, given the observations. Computation of the local potential can be visualised as passing a message from the observation to the node, see $m_Z$ in the left hand side of Figure 2. $\phi(x_i, x_j, \mathbf{z})$ is the pairwise clique potential. Pairwise potentials relate nodes $i$ and $j$ on either end of an edge. They transform the belief of node $j$ into a form suitable for node $i$ and vice versa. Lastly a product of all incoming messages is computed, *except* from the node to which the message is being sent. In the left hand side of Figure 2, the incoming messages are represented by $m_{a/c}$ and $m_{RT}$, while the outgoing messages $m_{ij}$ is visualised as $m_{out}$.

BP generates exact solution for graphs such as trees or polytrees. The proposed model is neither a tree nor a polytree. We therefore use Loopy Belief Propagation (LBP) [17], an approximation to BP. In LBP, message propagation continuous until the difference in messages falls below some threshold, or until a maximum number of iterations has been reached. Once message propagation is finished, the maximal configuration can be found by applying Equation 3 to each node in turn,

$$x_i^* \in \arg\max_{x_i} \left( \phi(x_i, \mathbf{z}) \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i) \right). \quad (3)$$

Here $x_i$ indicates the node while $x_i^*$ is its maximal configuration. As can be seen, maximisation is performed on a node's own belief (local potential) combined with all incoming messages. The maximal configuration for the node is then simply a state for which the combined belief is maximal.

## IV. JOINT MODEL FOR SCAN MATCHING AND CLUSTERING

### A. Model Definition

As discussed in Section II, solutions for motion detection/estimation and scan matching are generally defined independently. In reality though, both scan matching and moving object detection are strongly interdependent; laser points belonging to the same object will have similar associations and vice versa.

Following [23], motion detection is formulated as a clustering problem. The environment consists of one or more dynamic objects (one cluster for each dynamic object) together with a cluster representing the background. Laser points are clustered based on the object they belong to (i.e. those with similar motion patterns). Additionally, here we also include an *outlier* cluster for those points which can not reasonably be considered part of an object. Given the clusters for static and dynamic objects, motion estimation then determines the rotation and translation of each object.

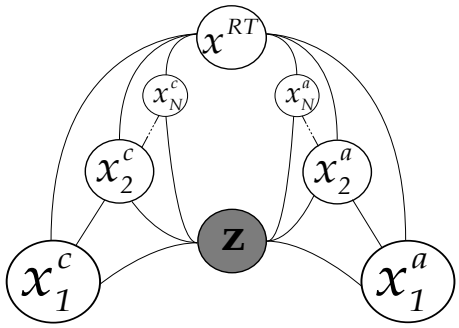


Fig. 1. Graphical model for combined scan matching, motion clustering and motion estimation.

A model (Equation 4), is proposed to solve moving object detection, motion estimation and data association simultaneously.

$$p(\mathbf{x} \mid \mathbf{z}) = p(\mathbf{x}^a, \mathbf{x}^c, x^{RT} \mid \mathbf{z}). \quad (4)$$

Equation 4 expresses a conditional distribution of three sets of hidden variables ($\mathbf{x}^a$, $\mathbf{x}^c$, $x^{RT}$) given an observation $\mathbf{z}$. The observation consists of two laser scans for which association and motion detection/estimation is to be determined.

The variables $\mathbf{x}^a$ consists of $N$ association nodes, where $N$ is the number of points in the first scan. Each of these nodes is discrete with $M+1$ states; $M$ being the number of points in the second scan. The states of $\mathbf{x}^a$ at node $i$ have the following interpretation: The first state indicates the likelihood that point $i$ in the first scan associates to point 1 in the second scan. The second state is the likelihood of association to the second point in the second scan, etc. Finally, the $M+1$ state represents the likelihood that point $i$ is an outlier.

The variables $\mathbf{x}^c$ consists of $N$ motion detection nodes, with $D$ discrete states each. Here $D$ represents the number of clusters a point can belong to. The interpretation is analogous to that of association. The first state represents the likelihood that point $i$ belongs to the first object (cluster), etc. By convention, the last state $D$ represents the cluster for outliers. The choice on number of clusters $D$ may be guided using *a priori* knowledge. If no such knowledge is available then $D$ can be set to $M+1$, i.e. start with a cluster for each point. Once the inference algorithm has clustered the points, any vacuous clusters can safely be discarded. The inference algorithm will assign probability mass to each node according to the likelihood of belonging to a cluster. Clusters that have no significant probability mass for all nodes are therefore not likely, and can be removed. For computational reasons it is advisable to reduce the number of clusters.

Lastly, the variable $x^{RT}$ represents the rotational $R$ and translational $T$ parameters of *each* object in the system except the outlier cluster. As such this is a multi-dimensional continuous variable. Note that the robot motion can be obtained from the rotation and translation parameters of the background object.

The graphical model corresponding to Equation 4 is shown in Figure 1. The model consists of two chains, one for motion clustering on the left, and one for data association on the right. The two chains are connected using a variable representing the motion estimates of the system, and a variable for the observations. The motivation for using a chain for both association and clustering stems from the way scan data is obtained. The laser scanner obtains data points sequentially and in a single plane; hence a chain to represent this acquisition. Note that any correlation in the data is not lost as there is a path from any one point in a chain to any other point.

### B. Inference

The inclusion of the variable $x^{RT}$ adds complexity. First and foremost, rotation and translation are continuous quantities whereas the nodes for association and clustering are discrete in nature. This could result in a mixing of sums and integrals in the inference procedure; in turn leading to difficulties in obtaining a solution. However, as explained in Section III-C, we formulate the problem as a MAP inference procedure defined as:

$$\mathbf{x}^{MAP} = \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{z}), \tag{5}$$

where $\mathbf{x} = \langle \mathbf{x}^a, \mathbf{x}^c, x^{RT} \rangle$. Max-product inference operates on the *maxima* of the hidden variable; it therefore does not suffer from any mixed sum integral problem - provided closed form solutions exist. The closed form solution to rotation and translation can efficiently be computed by minimising the error function in Equation 6 [13]:

$$R, T \leftarrow \arg\min_{R,T} \sum_{i=1}^{N} (z_{1,i} R + T - z_{2,x_i^a})^2, \tag{6}$$

where $z_{1,i}$ is the $i$-th point in the first scan and $z_{2,x_i^a}$ is the point in the second scan corresponding to the $i$-th association.

The message passing schedule is, in our case, dictated by the structure of the graph. In our model, the $\mathbf{x}^a$ and $\mathbf{x}^c$ nodes are indirectly linked through the motion estimate node, $x^{RT}$. This is done for practical reasons, as it allows inference to run as two chains simultaneously influencing each other. It does require a *flooding schedule* [11]; each node always sends messages to all of its neighbours. This ensures any change in one chain is always propagated to the other. In addition, the schedule visits $x^{RT}$ as every second node in the schedule to facilitate joint reasoning, i.e. a schedule such as: $x_1^c, x^{RT}, x_2^c, x^{RT}, ..., x_N^c, x^{RT}, x_1^a, x^{RT}, ..., x_N^a$.

In order to reduce the computational cost of sending messages to and from $x^{RT}$, the message passing algorithm is modified according to the right hand side of Figure 2. Our modified belief propagation algorithm treats the value for $x^{RT}$ *as if it was observed*. Outgoing messages $m_{out}$ contain the same information as long as we ensure no information (belief) is lost due to this change. In order to guarantee this, local features that depend on $x^{RT}$ are recomputed each time a node is visited. Any change to either rotation or translation is then incorporated into the outgoing message $m_{out}$. The

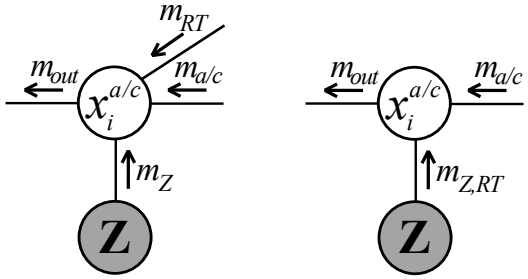


Fig. 2. Message passing for a single node. Left node shows standard belief propagation, right side uses proposed message propagation

observations $\mathbf{z}$ do not change, so any change in the message $m_{Z,RT}$ is solely due to changes in $x^{RT}$. In order for the modified algorithm to behave as if messages are being passed in both directions, the value of $x^{RT}$ needs to be updated inline with the message passing schedule. This requires $x^{RT}$ be recomputed after each node has been visited.

**Algorithm 1** Pseudo-code of the modified inference algorithm.

```
1:  x^RT ← InitialiseRT()
2:  for iteration = 1 to MaxIterations do
3:    for node = 1 to NumNodes do
4:      m_{Z,RT} ← ComputeLocalFeature(node, z, x^RT)
5:      for nb = 1 to Neighbours(node) do
6:        m_{a/c} ← CollectIncomingMessage(node, nb)
7:        m_out ← ConstructMessage(node, nb, m_{Z,RT}, m_{a/c})
8:        SendMessage(node, nb, m_out)
9:      end for
10:     x^RT ← ComputeRT()
11:   end for
12:   if CheckConvergence() = true then
13:     break
14:   end if
15: end for
16: x^RT ← ComputeRT()
17: for node = 1 to NumNodes do
18:   m_{Z,RT} ← ComputeLocalFeature(node, z, x^RT)
19:   m_{a/c} ← CollectIncomingMessage(node)
20:   x_{a/c} ← ComputeState(node, m_{Z,RT}, m_{a/c})
21: end for
```

The resulting algorithm is shown in Algorithm 1. To start, $x^{RT}$ is initialised on line 1 to reflect that it is 'observed'. Belief propagation is performed on lines 2-15. Before a node propagates belief to one of its neighbours it computes its own belief (line 4) from the observations $\mathbf{z}$ and the value for $x^{RT}$. For each of its neighbours Equation 2 is computed (lines 5-9) and the output message is sent to its neighbour. After sending messages to all neighbours, the crucial step of updating $x^{RT}$ is executed on line 10; minimising Equation 6. After each iteration of the algorithm convergence is checked, only if the algorithm has converged can it be terminated early. Finally, lines 16-21 compute the state of each node analogous to standard LBP inference (Equation 3).

### C. Feature Description

CRFs owe much of their popularity to the feature functions $\mathbf{f}_c$ in Equation 1. These encapsulate domain specific knowledge which can subsequently be used in the probabilistic framework of the CRF. For completeness a brief description of each feature is provided below. The reader is referred to [19], [23] for more detailed descriptions.

*1) Association Local Features:* A first class of association local features use geometric properties of the data to identify patterns (shapes) around a single point, say point $i$, in the first scan. They then try to find the same pattern around *each* point in the second scan. The resulting difference is an error metric; points in the second scan that have a similar pattern (a small value for the error metric) are more likely candidates for association to point $i$ in the first scan. The pattern finding features are:

- *Distance*: Uses the distance to its first, third or fifth neighbour.
- *Angular*: Uses the angle between the first, third or fifth neighbours on either side.
- *Geodesic*: Uses the accumulated distance to its first, third or fifth neighbour; visiting all points in between.
- *Radius*: Uses the range value.

In addition to the shape based features above, two other features are used. These use the structure of the data and optionally contextual information.

- *Translation*: Uses the distance between a point in the first scan with every point in the second scan - analogous to ICP.
- *Registered Translation*: Transforms each point in the first scan according to its motion estimate. Then uses the distance between the transformed point and every point in the second scan.

The above features are not directly used in Equation 1. They are, instead, used as inputs to boosting [7], [19]. The outputs of boosting (the experiments employ AdaBoost [6] with 50 decision stumps) are used as local feature values in Equation 1. Using boosting results in better estimates for the local potentials. There are two boosting features used for association: 1) *Data Boosting* computes the likely associations and 2) *Outlier Boosting* determines the likelihood that a point is an outlier.

*2) Association Pairwise Features:* These features fall roughly into two categories. Those that use the structure of the scan acquisition and those that use the observations.

The first category of features are those that use scan acquisition structure. In an ideal world, without noisy measurements and outliers, it would be straight forward to relate the association of node $i$ with that of node $i+1$. If node $i$ associates to point $j$ then node $i+1$ can reasonably be expected to associate to point $j+1$.

- *Sequence*: Expresses the sequential nature of association by an identity matrix with the diagonal shifted.
- *Pairwise Outlier*: Expresses how outliers impact on association transitions - from inlier to outlier and vice versa.

The second category of pairwise features operate similarly to the pattern finding local features. They use a measure between two points on an edge in the first scan and compare this measure with *all* possible combinations of this measure in the second scan. The comparison produces a metric of how pairs of points are related between the two scans (i.e. a transition).

- *Pairwise Distance*: Uses the distance between points on either end of an edge.
- *Pairwise Translation*: Uses the vector between two points corresponding to an edge.

*3) Clustering Local Features:* The purpose of these features is to determine whether points are part of the static background, part of a dynamic object or are outliers.

- *Cluster Distance*: Uses distance between two scans to cluster points based on their motion.
- *Outlier*: Uses a threshold value to indicate if the point is an outlier.
- *Cluster Inlier* Enforces inlier consistency between association and clustering.

*4) Clustering Pairwise Features:* With the exception of the outlier feature, all these features cluster by incorporating neighbourhood information.

- *Sequence*: Enforces if neighbouring points belong to the same cluster; an identity matrix.
- *Neighbour Weight*: Captures (non-)neighbouring relationships; scaled by the distance between the points.
- *Neighbour Stiffness*: Captures (non-)neighbouring relationships; scaled by the distance between associated pairs of neighbouring points.
- *Neighbour Distance*: Expresses that points belonging to the same cluster should preserve their relative distance after transformation.
- *Pairwise Outlier*: Expresses how outliers impact on cluster transitions - from inlier to outlier and vice versa.

## V. EXPERIMENTS

We perform experiments using data collected with a laser scanner mounted on a car travelling around the University of Sydney campus [23]. 31 pairs of scans containing both static and dynamic objects were manually annotated. The 31 scan pairs can be divided into 8 subsets; each subset consisting of sequential scan pairs. A single scan contains on average about 300 points. As a result the proposed model consists of a graph of 600 nodes (300 association + 300 clustering). The clustering nodes are initialised with 5 states for each node while the association nodes will have on average 301 states.

The proposed joint model is compared to three techniques that compute laser point association and motion clustering as separate tasks. The first technique employs ICP and the output associations provide translation and rotation parameters from K-Means clustering [15]. The second technique uses CRF-Matching [19] instead of ICP for association and again K-Means for clustering. Note that K-Means requires the number of clusters to be defined in advance. In the experiments the number of clusters is set to 4 - the maximum

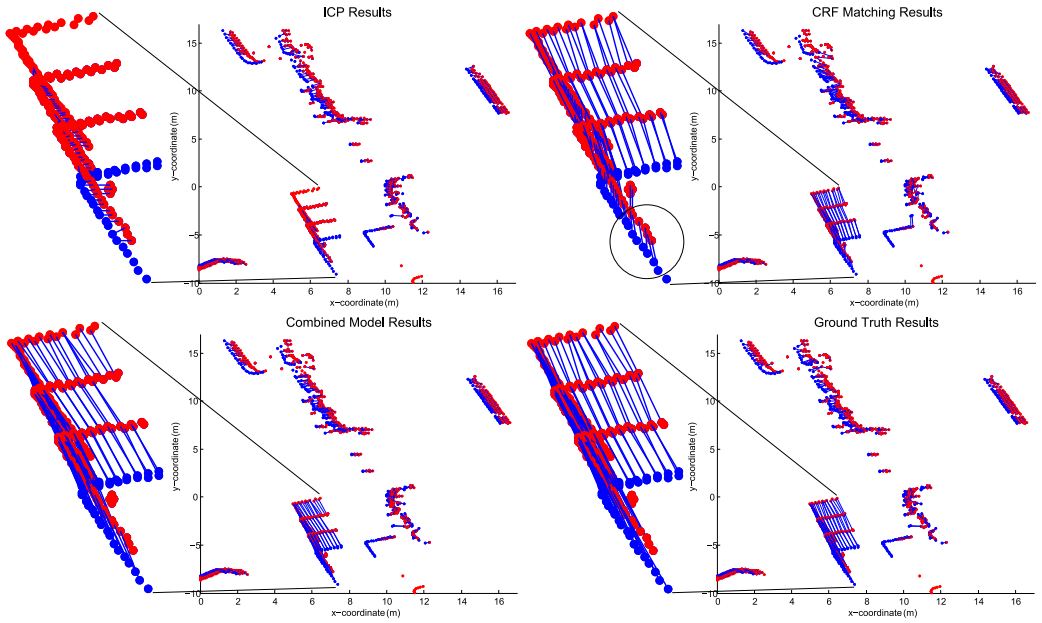


Fig. 3. A sequence of 4 scan association results with magnified dynamic object associations. Top row, left to right: ICP, CRF-Matching. Bottom row, left to right: Proposed Joint Model, Ground Truth. Lines indicate MAP associations between laser points (outlier points have no connecting lines). All techniques perform well on the background. Looking at the dynamic object, ICP is unable to associate the dynamic object. CRF-Matching does reasonably well but struggles in the encircled area. The proposed model correctly associates points in the direction of motion.

| Technique | Accuracy(%) | SED | Outlier(%) (true/ratio) |
|---|---|---|---|
| ICP | 69.19 | 119.10 | 34.60 (11.07/89.79) |
| CRF-Matching | 71.03 | 116.16 | 13.01 (11.07/57.93) |
| Combined Model | 74.13 | 96.13 | 12.96 (11.07/73.37) |

TABLE I
ASSOCIATION METRICS FOR THE DIFFERENT TECHNIQUES.

based on the ground truth. Finally, the third technique uses CRF-Matching and CRF-Clustering in turn. CRF-Clustering is essentially the clustering layer in Figure 1 while CRF-Matching is the association layer. This technique equates to removing all the links connecting the two chains through $x^{RT}$ in Figure 1 - this allows for a clear comparison and shows the advantage of joint reasoning about motion detection and scan matching.

Experiments are conducted in a leave-one-out cross validation fashion. All CRF based techniques require learning of the weights (18/12/30 weights for CRF-Matching/CRF-Clustering/proposed model respectively). As such, 30 scan pairs are used to training the model and 1 scan pair is used for testing. This process is then repeated for each of the 31 scan pairs and averaging the results. This scheme ensures that the results represent the model's ability to deal with each of the scan pairs individually. ICP and K-Means do not require training, for these techniques the average over the 31 scans is computed for fair comparison.

Figure 3 shows examples of association results for each of the different association techniques together with the association result of the proposed model and ground truth. The figures contain 4 sequential scans, with the blue scan being the most recently acquired scan. The dynamic object is magnified in all figures. Looking at the results it is clear that all perform quite well on the static points. The difference is how the different techniques deal with the dynamic object. ICP does poorly, this is to be expected since ICP uses only nearest neighbour information for determining associations. CRF-Matching does significantly better due to the fact that it performs associations based on local shape information. It is, however, unable to correctly associate the points in the lower section (circled) of the dynamic object as the local shape of these points is similar. The proposed model on the other hand has clustering information available. This allows it to correctly associate in the direction of motion.

The output of motion detection is analogous. Figure 4 shows the results for a single scan pair - for clarity. As can be seen, the results for K-Means clustering depend very much on the quality of the associations. Using ICP for the associations, the results are inconsistent. K-Means with CRF-Matching does significantly better but is adversely affected by K-Means requiring the number of clusters to be determined in advance. CRF-Clustering does better, because like CRF-Matching, CRF-Clustering is able to infer more from the scans. The proposed model does exceptionally well. It correctly determines the number of clusters in the data and there are very few mis-classifications - only a few in the corner of the dynamic object.

In addition to the visual comparison, a quantitative measure of similarity between the association/clustering results and the ground-truth is given. Tables I and II present a performance comparison in terms of accuracy, String Edit Distance (SED) [21] and outlier detection. The accuracy is

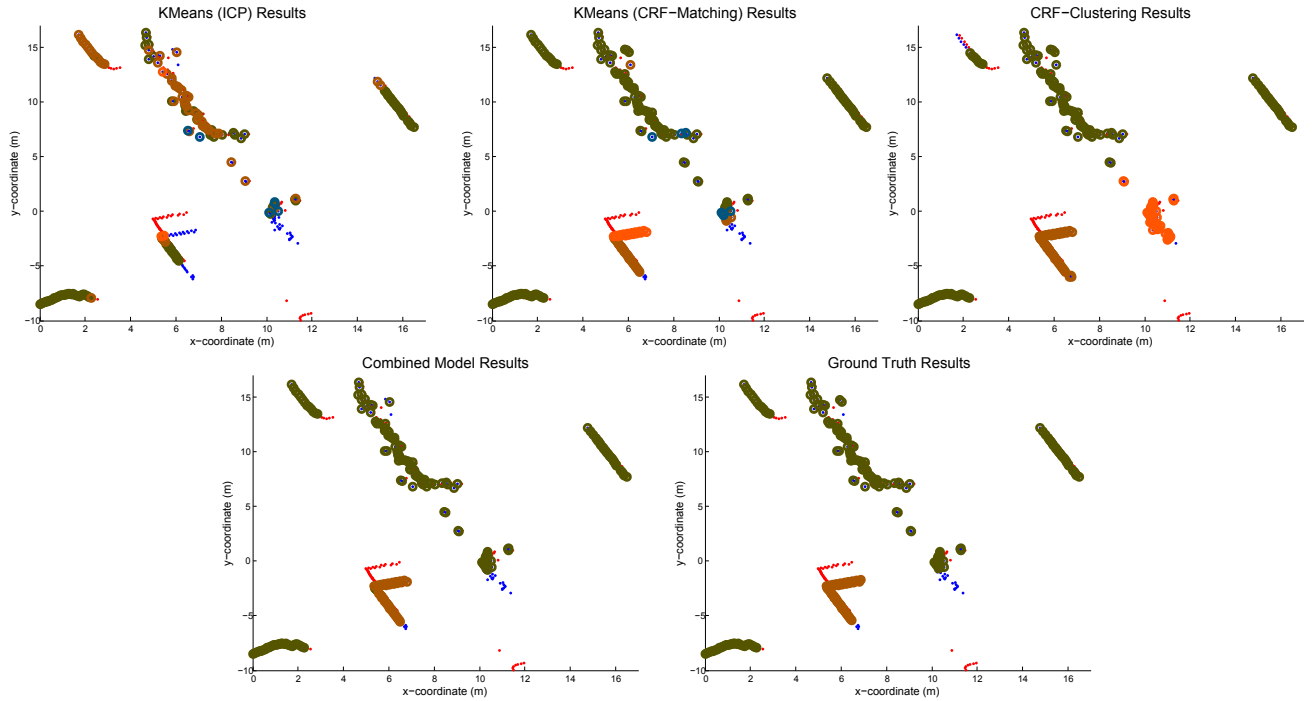


Fig. 4. Motion detection results. Top row, left to right; K-Means initialised with ICP, K-Means initialised with CRF-Matching, CRF-Clustering. Bottom row, left to right: Proposed Joint Model, Ground Truth. Different colours indicate different clusters (dark green for the static background), outlier points have no coloured circle. K-Means (ICP) has very poor results mainly due the poor associations. CRF-Clustering and K-Means (CRF-Matching) have similar results, though CRF-Clustering results are a little more consistent. The proposed model has near perfect result.

| Technique | Accuracy(%) | SED | Outlier(%) (true/ratio) |
|---|---|---|---|
| K-Means (ICP) | 43.16 | 243.87 | 34.60 (11.07/89.79) |
| K-Means (CRF) | 60.42 | 215.55 | 13.01 (11.07/57.93) |
| CRF-Clustering | 60.04 | 190.03 | 24.35 (11.07/55.64) |
| Combined Model | 83.22 | 79.61 | 13.12 (11.07/73.37) |

TABLE II
MOTION DETECTION METRICS FOR THE TECHNIQUES.

| Technique | Rotation Error (radians) | Translation Error (m) |
|---|---|---|
| K-Means (ICP) | 0.02 | 1.01 |
| K-Means (CRF) | 0.04 | 0.77 |
| CRF-Clustering | 0.05 | 0.88 |
| Combined Model | 0.03 | 0.56 |

TABLE III
MOTION ESTIMATION METRICS FOR THE TECHNIQUES.

the percentage of correctly associated/clustered points[1]. The SED is a measure of how many permutations are required on a string to make it match another. Here the two strings are the association/clustering results on the one hand and the ground-truth on the other. Therefore, the less the SED the better. Finally the percentage of outliers gives a measure of how useful the solution is. Solutions with a high percentage of outliers may give a reasonable accuracy/SED but they are not practical. Outliers are presented as the percentage of outliers found, together with the true percentage of outliers (from the ground-truth), and a ratio (percentage) of ground-truth outliers that were found.

Table I confirms the visual results of Figure 3 for the complete data set. The combined model outperforms both ICP and CRF-Matching. A note on the results of ICP. Based on the metrics provided, accuracy & SED, one might be inclined to conclude that ICP did rather well. To a certain extent this is true; the different scans were reasonably closely aligned and there are relatively few dynamic points (ratio of dynamic over static points is $\sim 0.14$). ICP's solution however, consists (on average) of more than a third outliers. It should have found around 11 percent of outliers. This high percentage of outliers subsequently results in poor estimates for motion detection. Metric comparison for the motion parameters is given in Tables II and III. The comparison shows again that joint reasoning about association and motion produces better results. Table III shows average results for correctly clustered objects only. The relatively good performance of the K-Means based techniques is primarily due to correct clustering of the background.

[1]A good association accuracy metric is difficult to define; associations near to the true associations require a different penalty from those further away.

## VI. CONCLUSIONS AND FUTURE WORKS

This paper presented a probabilistic model that jointly reasons about laser point association, motion clustering and motion estimation. The experimental results have demonstrated that much can be gained by modelling different (but dependent) problems in the same statistical framework. We

believe the methods presented here are an initial step towards integrating multiple tasks in mobile robotics.

In its present form, the clustering and association chain models are (indirectly) linked using the rotation and translation node. In future these two models will be directly connected thus forming a clustering and association lattice. The lattice more closely represents the dependencies between the two models. The drawback of this is an increment in the computational complexity. Note that a different graph structure also allows for different message passing schedules with different performance characteristics (see comment on performance next).

Inference in the combined model is slow. For laser scans with 361 points, this can take up to 4 minutes in our Matlab implementation - depending on convergence. Analysis has shown 3 performance bottlenecks. First, some of the pairwise features are poorly implemented. This is exacerbated by the second bottleneck; the fact that a flooding schedule is used for message passing. Lastly the cost of computing Equation 6. There are, however, several alternatives to speed up inference. One possibility is to constrain the number of states in the association model. For example, a particular laser point can only be associated to one of its 10 nearest neighbours rather than to all 361 points of the other scan. This would significantly reduce the cost of searching for an association. As well as a reduction in the computational cost of message propagation - the size of the pairwise feature functions will be quadratically reduced. Additionally, we plan to investigate better inference algorithms, especially ones with strong convergence guarantees. In this case, Linear Programming (LP) relaxations appear as a potential candidate.

## ACKNOWLEDGEMENTS

This work has been supported by the Rio Tinto Centre for Mine Automation and the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

## REFERENCES

[1] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[2] Y. Bar-Shalom and X.-R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, 1995.

[3] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24, 1975.

[4] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun. Towards object mapping in dynamic environments with mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[5] M. Bosse and R. Zlot. Continuous 3d scan-matching with a spinning 2d laser. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2009.

[6] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Proc. of the International Conference on Machine Learning (ICML)*, 1996.

[7] S. Friedman, D. Fox, and H. Pasula. Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[8] S. Granger and X. Pennee. Multi-scale EM-ICP: A fast and robust approach for surface registration, 2002. Internal research report, INRIA.

[9] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2003.

[10] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems (NIPS 2008)*, 2008.

[11] F. Kschischang, B. J. Frey, and H. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

[12] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning (ICML)*, 2001.

[13] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 1994.

[14] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18, 1997.

[15] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[16] L. Montesano, J. Minguez, and L. Montano. Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2005.

[17] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 1999.

[18] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

[19] F. Ramos, D. Fox, and H. Durrant-Whyte. CRF-matching: Conditional random fields for feature-based scan matching. In *Proc. of Robotics: Science and Systems*, 2007.

[20] D. Rodriguez-Losada and J. Minguez. Improved data association for icp-based scan matching in noisy and dynamic environments. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2007.

[21] D. Sankoff and J. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, 1983.

[22] D. Schulz, W. Burgard, D. Fox, and A. B Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[23] G. Tipaldi and F. Ramos. Motion clustering and estimation with conditional random fields. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

[24] C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[25] C. Wang, C. Thorpe, M. Hebert, S. Thrun, and H. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(6), June 2007.

[26] D. Wolf and G. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19:53–65, 2005.