Data Association in Stochastic Mapping using the Joint Compatibility Test

José Neira, Juan D. Tardós

***Abstract*— In this work we address the problem of robust data association for simultaneous vehicle localization and map building. We show that the classical gated nearest neighbor approach, which considers each matching between sensor observations and features independently, ignores the fact that measurement prediction errors are correlated. This leads to easily accepting incorrect matchings when clutter or vehicle error increase. We propose a new measurement of the joint compatibility of a set of pairings that successfully rejects spurious matchings. We show experimentally that this restrictive criterion can be used to efficiently search for the best solution to data association. Unlike the nearest neighbor, this method provides a robust solution in complex situations, such as cluttered environments or when revisiting previously mapped regions.**

***Index Terms*— Stochastic Mapping, Data Association, Mahalanobis Distance, Nearest Neighbor**

I. Introduction

The use of a sensor mounted on a vehicle to build and update a map of the environment where the vehicle is navigating poses a specially daunting data association problem. Data association consists in relating sensor observations with the elements included in the map. Obtaining a correct solution is crucial, because a misassignment causes location estimation methods, such as the Extended Kalman Filter (EKF), to diverge [1].

Data association may be posed as a search problem in the space of observation-feature correspondences [2]. The complexity of finding correspondences between a set of m sensor measurements and n map features is exponential on the number of measurements: if there are $n_i + 1$ possible pairings for the i-th measurement (allowing repetitions and including the possibility that it be spurious), the correct hypothesis is to be found among an exponential space of $\prod_{i=1}^{m}(n_i + 1)$ alternatives. Two main factors determine the size of the solution space:

1) *Clutter*: both m and n_i are proportional to the density of features in the environment.
2) *Imprecision*: n_i also grows with the imprecision of the vehicle location, and of the sensor being used.

A data association algorithm is composed of two elements: a *test* to determine the compatibility between a sensor observation and a map feature given an estimation of the vehicle location, and a *selection criterion* to choose the best matchings among the set of compatible matchings.

In stochastic mapping [3], this problem is frequently addressed using the gated nearest neighbor (NN) algorithm, a classical technique in tracking problems [1]. The normalized squared innovation test is used to determine compatibility, and then the nearest neighbor rule (smallest Mahalanobis distance) is used to select the best matchings. This technique, sometimes along with additional heuristics, such as accepting matchings only for observations with a single candidate map feature, has been used by many authors [4], [5], [6], [7].

The great advantage of this solution, apart from its conceptual simplicity, is its $O(mn)$ computational complexity. However, a very important fact is being overlooked: the innovations in the matchings of different observations obtained from the same vehicle position are *correlated*. As we show in section III, this causes the individual innovation compatibility test to be too permissive: it easily accepts hypotheses formed by mutually inconsistent pairings, which leads to divergence in the estimation of the state.

The power of this test to detect spurious matchings decreases as vehicle imprecision or clutter grow. NN is reliable for features such as segmented walls from laser sensors, where clutter is low and sensor precision is high, as long as vehicle error is moderate [5]. However, its reliability quickly plummets as the uncertainty of features relative to the vehicle increases, as is always the case when revisiting previously mapped regions after a long loop. Reliability also plummets when using less precise sensors, such as sonar [8], or edge-based monocular vision [9].

More robust algorithms have been proposed, such as multi-tracking, which obtains hypotheses where *temporal* coherence is guaranteed. Multi-tracking actually increases complexity, since it is necessary to maintain one map per hypothesis [10], [11]. Sensor specific solutions, such as visual tracking [12], can perform *local* data association very efficiently, but cannot be used to solve the crucial revisiting problem. Alternative approaches search for the best solution in the vehicle pose space rather than in the correspondence space [13], [14], [15].

In other approaches, geometric constraints between features are used to obtain hypotheses with pairwise compatible pairings. Baley et. al. [16] consider relative distances and angles between points and lines in two laser scans, and use a graph theoretic approach to find the largest number of pairwise compatible pairings. Castellanos and Tardós [17] also use binary constraints to localize the robot with an *a priori* map using an interpretation tree approach. However, pairwise compatibility does not guarantee *joint* compatibility [2], and additional validations are required.

The material in this paper was partially presented at IEEE Int. Conf. Robot. Automat., Workshop W4: Mobile Robot Navigation and Mapping, San Francisco, CA, April 2000.

The authors are with the Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza: c/María de Luna 3, E-50015 Zaragoza, SPAIN. email: {jneira,tardos}@posta.unizar.es.

This work was partially supported by Spanish Dirección General de Investigación, project DPI2000-1265.

In this work we define a criterion that, taking explicitly into account the correlations between the innovations, determines the joint compatibility of a set of pairings. We show that this criterion is much more restrictive than the individual innovation test, and that it can be efficiently computed. The use of this criterion in a branch and bound (BB) search algorithm results in a very robust solution to data association, with an efficient traversal of the solution space. We compare both the robustness and efficiency of the NN and BB algorithms in an experiment with a Labmate mobile robot equipped with a trinocular vision system, building a map along a loop trajectory of $60m$. We show experimentally that in simple situations both algorithms are robust, with an equivalent performance. When there is an increase in clutter and/or imprecision, NN breaks down, while BB maintains its robustness, with a computational cost that is acceptable for real time applications.

II. The Classical Nearest Neighbor Approach

A. Problem definition

In stochastic mapping [3], [6], the state of a vehicle R and of a set of n features $\{F_1, \cdots, F_n\}$ of the environment where the vehicle is navigating is represented by a vector $\mathbf{x}$. Let $\hat{\mathbf{x}}$ be the estimation of the vehicle and feature locations, and $\mathbf{P}$ the covariance of the estimation error:

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_R \\ \hat{\mathbf{x}}_{F_1} \\ \vdots \\ \hat{\mathbf{x}}_{F_n} \end{pmatrix}; \quad \mathbf{P} = \begin{pmatrix} \mathbf{P}_R & \mathbf{P}_{RF_1} & \cdots & \mathbf{P}_{RF_n} \\ \mathbf{P}_{RF_1}^T & \mathbf{P}_{F_1} & \cdots & \mathbf{P}_{F_1F_n} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{P}_{RF_n}^T & \mathbf{P}_{F_1F_n}^T & \cdots & \mathbf{P}_{F_n} \end{pmatrix}$$

In a similar way, let $\hat{\mathbf{y}}$ represent a set of m measurements $\{E_1, \cdots, E_m\}$ of environment features, obtained using a sensor mounted on the vehicle, affected by white Gaussian noise:

$$\hat{\mathbf{y}} = \mathbf{y} + \mathbf{u}; \quad \mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{S})$$

$$\hat{\mathbf{y}} = \begin{pmatrix} \hat{\mathbf{y}}_{E_1} \\ \vdots \\ \hat{\mathbf{y}}_{E_m} \end{pmatrix}; \quad \mathbf{S} = \begin{pmatrix} \mathbf{S}_{E_1} & \cdots & \mathbf{S}_{E_1E_m} \\ \vdots & \vdots & \vdots \\ \mathbf{S}_{E_1E_m}^T & \cdots & \mathbf{S}_{E_m} \end{pmatrix}$$

where $\mathbf{y}$ is the theoretical value of the observations. A measurement E_i and its corresponding feature F_{j_i} are related by an *implicit measurement function* [18] of the form:

$$\mathbf{f}_{ij_i}(\mathbf{x}, \mathbf{y}) = \mathbf{0} \tag{1}$$

which states that the relative location between the measurement and the corresponding feature must be zero.

The purpose of a data association algorithm is to generate a hypothesis $\mathcal{H}_m = \{j_1, \cdots, j_m\}$, that pairs each measurement E_i with a map feature F_{j_i} (when $j_i = 0$, the measurement is considered spurious). This exponential solution space can be represented as an *interpretation tree* of m levels [2]; each node at level i, called an *i-interpretation*, provides an interpretation for the first i measurements. Each node has $n_i + 1$ branches, corresponding to each of the alternative interpretations for measurement E_i (including the possibility that the measurement be spurious and allowing map feature repetitions in the same hypothesis). Data association algorithms must select in some way one of the $\prod_{i=1}^{m}(n_i + 1)$ m-interpretations as the correct hypothesis, carrying out validations to determine the compatibility between sensor measurements and map features. Once a hypothesis has been obtained, it can be used to improve the estimation of $\mathbf{x}$.

B. Individual Compatibility Nearest Neighbor

The Individual Compatibility Nearest Neighbor (ICNN) simply pairs each measurement with the feature considered most compatible according to eq. (1). Since usually the measurement function is non-linear, linearization around the current estimation is necessary:

$$\mathbf{f}_{ij_i}(\mathbf{x}, \mathbf{y}) \simeq \mathbf{h}_{ij_i} + \mathbf{H}_{ij_i}(\mathbf{x} - \hat{\mathbf{x}}) + \mathbf{G}_{ij_i}(\mathbf{y} - \hat{\mathbf{y}}) \tag{2}$$

where:

$$\mathbf{h}_{ij_i} = \mathbf{f}_{ij_i}(\hat{\mathbf{x}}, \hat{\mathbf{y}}); \quad \mathbf{H}_{ij_i} = \left.\frac{\partial \mathbf{f}_{ij_i}}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}})}; \quad \mathbf{G}_{ij_i} = \left.\frac{\partial \mathbf{f}_{ij_i}}{\partial \mathbf{y}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}})}$$

Vector $\mathbf{h}_{ij_i}$ represents the innovation of the pairing between E_i and F_{j_i}. From eqs. (1) and (2) its covariance can be obtained as:

$$\begin{aligned} \mathbf{C}_{ij_i} &= \mathbf{H}_{ij_i} Cov(\mathbf{x} - \hat{\mathbf{x}}) \mathbf{H}_{ij_i}^T + \mathbf{G}_{ij_i} Cov(\mathbf{y} - \hat{\mathbf{y}}) \mathbf{G}_{ij_i}^T \\ &= \mathbf{H}_{ij_i} \mathbf{P} \mathbf{H}_{ij_i}^T + \mathbf{G}_{ij_i} \mathbf{S} \mathbf{G}_{ij_i}^T \end{aligned} \tag{3}$$

The individual compatibility (IC) between E_i and F_{j_i} can be determined using an innovation test that measures the Mahalanobis distance as follows:

$$D_{ij_i}^2 = \mathbf{h}_{ij_i}^T \mathbf{C}_{ij_i}^{-1} \mathbf{h}_{ij_i} < \chi_{d,\alpha}^2 \tag{4}$$

where $d = \dim(\mathbf{f}_{ij_i})$ and α is the desired confidence level. This test, applied to the predicted state, determines the subset of map features that are compatible with a measurement E_i (and thus the value of n_i).

The nearest neighbor selection criterion for a given measurement consists in choosing among the features that satisfy eq. (4), the one with the smallest Mahalanobis distance. This algorithm is frequently used given its conceptual simplicity and computational efficiency: it performs mn compatibility tests, making it linear with the size of the map. For a large $\mathbf{C}_{ij_i}$, its inversion may be a costly operation. Several methods to calculate lower bounds for $D_{ij_i}^2$ have been proposed, which in some cases allow to reject the pairing between E_i and F_{j_i} without inverting $\mathbf{C}_{ij_i}$ [19], [20], [21].

C. Limitations of the Nearest Neighbor

To illustrate the limitations of this approach, consider the simple example of a robot R that traverses a monodimensional space, where there are two features (fig. 1). Assume that this robot is equipped with a sensor capable of detecting these features. At step 1, the robot observes the two features,

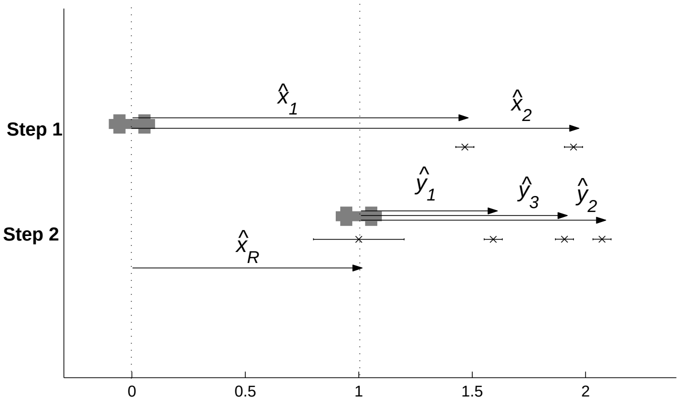


Fig. 1. Monodimensional robot with two features. Step 1: map creation from two sensor measurements taken from initial position. Step 2: data association problem after $1m$ motion with three sensor measurements, $\hat{y}_3$ spurious. Intervals represent 2σ uncertainty bounds.

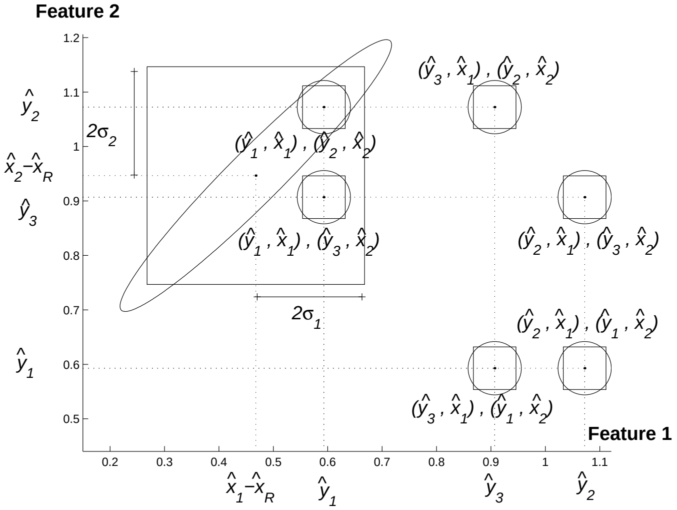


Fig. 2. Simulation of a monodimensional robot. Each axis represents the predicted location of a feature with respect to the robot location ($\hat{x}_1 - \hat{x}_R$ and $\hat{x}_2 - \hat{x}_R$ respectively) together with the three actual measurements $\hat{y}_1$, $\hat{y}_2$ and $\hat{y}_3$. The big square region represents the uncertainty of each prediction considered independently, while the ellipsoidal region represents this uncertainty considering the correlation between the predictions. Each small group of a square and a circle represents an alternative matching hypothesis, with its measurement uncertainty.

and they are included in the stochastic map. At step 2 the robot moves $1m$, and obtains three measurements: $\hat{y}_1$, $\hat{y}_2$, and $\hat{y}_3$ (spurious). The uncertainties in robot motion and sensor measurements are $\sigma_R = 0.1$ and $\sigma_S = 0.02$. In this case we have:

$$\begin{aligned} \mathbf{f}_{ij}(\mathbf{x}, \mathbf{y}) &= x_R + y_i - x_j = 0 \\ \mathbf{h}_{ij} &= \hat{x}_R + \hat{y}_i - \hat{x}_j \end{aligned}$$

In fig. 2, the big square represents the 2σ acceptance region of IC, and the six small squares represent the uncertainty of the six main pairing hypotheses (we consider here the additional restriction that two measurements cannot come from the same feature). According to IC, only measurement $\hat{y}_1$ is compatible with feature $\hat{x}_1$ (horizontal axis in fig. 2), while both $\hat{y}_2$ and $\hat{y}_3$ are compatible with $\hat{x}_2$ (vertical axis). Thus IC accepts the two possible hypotheses $\{(\hat{y}_1, \hat{x}_1), (\hat{y}_2, \hat{x}_2)\}$ and $\{(\hat{y}_1, \hat{x}_1), (\hat{y}_3, \hat{x}_2)\}$ because their squares intersect the square region of acceptance. As $\hat{y}_3$ lies closer to $\hat{x}_2 - \hat{x}_R$ than $\hat{y}_2$, NN would prefer the second hypothesis, pairing $\hat{y}_3$ with $\hat{x}_2$ incorrectly.

This occurs because IC considers *individual* compatibility between a measurement and a feature. However, individually compatible pairings are not guaranteed to be *jointly* compatible to form a consistent hypothesis. Even if sensor measurements are independent, correlations in the stochastic map are always present and cannot be ignored [5]. Furthermore, the predicted measurements are always correlated because they are affected by the same robot position error. Graphically this means that, for a given confidence level, the region of acceptance of the predicted measurements is an *ellipsoidal region* instead of a square region (fig. 2). From this perspective, in our example it is clear that pairings $\{(\hat{y}_1, \hat{x}_1), (\hat{y}_3, \hat{x}_2)\}$ are not simultaneously acceptable: the region of uncertainty of this hypothesis (circular, given that the precision of this sensor has been considered constant) does not intersect the ellipsoidal region of acceptance of the predicted measurements. Only the region corresponding to hypothesis $\{(\hat{y}_1, \hat{x}_1), (\hat{y}_2, \hat{x}_2)\}$ does.

This simple example shows that with ICNN there is a high risk of obtaining an inconsistent hypothesis and thus updating the state vector with a set of incompatible measurements, which will cause EKF to diverge. As vehicle error grows with respect to sensor error, the ratio between the area of the square and the area of the ellipse grows (due to the increase in the correlation between the vehicle and the predicted measurements), and thus the discriminant power of IC decreases. For this reason, the ICNN algorithm is adequate only when these two conditions are satisfied:

1) The robot error is smaller than the distance between the features, so that it is unlikely that two features pass the IC test for the same observation.
2) Spuriousness is low enough so that it is unlikely that a spurious measurement will fall inside the acceptance region of some feature.

III. Obtaining consistent hypotheses

A. Sequential Compatibility Nearest Neighbor

A simple way to assure that the resulting hypothesis contains jointly compatible pairings is to use a Sequential Compatibility Nearest Neighbor (SCNN) algorithm. Sequential Compatibility (SC) is an innovation test that determines compatible features for a measurement E_i, given that a hypothesis $\mathcal{H}_{i-1} = \{j_1, \cdots, j_{i-1}\}$, corresponding to pairings $\{(E_1, F_{j_1}), \cdots, (E_{i-1}, F_{j_{i-1}})\}$ has been established and has been used to obtain an estimation $\hat{\mathbf{x}}_{i-1}$ of the state. If the Mahalanobis distance $D^2_{ij_i}$, corresponding to a feature F_{j_i}, satisfies eq. (4) given $(\hat{\mathbf{x}}_{i-1}, \mathbf{P}_{i-1})$, the pairing (E_i, F_{j_i}) will be consistent with *all* pairings in $\mathcal{H}_{i-1}$. If this feature is considered the nearest compatible with measurement E_i, the measurement can be used to obtain the new estimate $\hat{\mathbf{x}}_i$ of the state vector and its covariance $\mathbf{P}_i$. In the next iteration, a pairing for measurement E_{i+1} will be determined applying eqs. (2) and (4) with the updated state estimation $(\hat{\mathbf{x}}_i,\, \mathbf{P}_i)$.

SCNN is an $O(mn) + O(mn^2)$ algorithm, quadratic with the size of the map: for each of the m measurements, SCNN requires evaluating compatibility with the n map features, $O(n)$, and updating the stochastic map, $O(n^2)$ (recent works address the reduction of this complexity [22], [23]). It is appealing because it guarantees that all pairings belonging to the resulting hypothesis are jointly compatible. However, it ignores the fact that these pairings may anyway be *incorrect*: a feature may be compatible with an unrelated or spurious sensor measurement just by chance. In our example, if SCNN tries to pair observation $\hat{y}_3$ in the first step, it will pair it incorrectly with $\hat{x}_2$ (fig. 2). The pairing will be used to re-estimate the stochastic map, and no more pairings will be acceptable in subsequent steps. This risk increases with clutter, and robot or sensor error. In this greedy algorithm, the decision to pair a measurement with its most compatible feature is never reconsidered, and thus spurious pairings may be included in the hypothesis and integrated in the state estimation, specially during the initial iterations. This leads again to a reduction in uncertainty with no reduction in error, i.e. inconsistency.

B. Joint Compatibility Branch and Bound

Reconsideration of the established pairings is necessary to limit the possibility of accepting a spurious pairing. The probability that a spurious pairing is jointly compatible with all the pairings of a given hypothesis decreases as the number of pairings in the hypothesis increases. For this reason, we require a search algorithm to traverse the interpretation tree in search for the hypothesis that includes the largest number of *jointly compatible* pairings. SC could be used to restrict the search to tree nodes representing hypotheses with jointly compatible pairings, but it requires the update of the whole stochastic map for each pairing considered, $O(n^2)$, which can be computationally very expensive in a search algorithm.

We use an alternative formulation to establish the consistency of a hypothesis $\mathcal{H}_i = \{j_1, \cdots, j_i\}$, called Joint Compatibility (JC), which uses a joint implicit function $\mathbf{f}_{\mathcal{H}_i}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, where:

$$\begin{aligned}
\mathbf{f}_{\mathcal{H}_i}(\mathbf{x}, \mathbf{y}) &= \begin{pmatrix} \mathbf{f}_{\mathcal{H}_{i-1}}(\mathbf{x}, \mathbf{y}) \\ \mathbf{f}_{ij_i}(\mathbf{x}, \mathbf{y}) \end{pmatrix} = \begin{pmatrix} \mathbf{f}_{1j_1}(\mathbf{x}, \mathbf{y}) \\ \vdots \\ \mathbf{f}_{ij_i}(\mathbf{x}, \mathbf{y}) \end{pmatrix} \\
&\simeq \mathbf{h}_{\mathcal{H}_i} + \mathbf{H}_{\mathcal{H}_i}(\mathbf{x} - \hat{\mathbf{x}}) + \mathbf{G}_{\mathcal{H}_i}(\mathbf{y} - \hat{\mathbf{y}}) \qquad (5)
\end{aligned}$$

$$\mathbf{h}_{\mathcal{H}_i} = \mathbf{f}_{\mathcal{H}_i}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = \begin{pmatrix} \mathbf{h}_{\mathcal{H}_{i-1}} \\ \mathbf{h}_{ij_i} \end{pmatrix} = \begin{pmatrix} \mathbf{h}_{1j_1} \\ \vdots \\ \mathbf{h}_{ij_i} \end{pmatrix} \qquad (6)$$

$$\mathbf{H}_{\mathcal{H}_i} = \left.\frac{\partial \mathbf{f}_{\mathcal{H}_i}}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}})} = \begin{pmatrix} \mathbf{H}_{\mathcal{H}_{i-1}} \\ \mathbf{H}_{ij_i} \end{pmatrix} = \begin{pmatrix} \mathbf{H}_{1j_1} \\ \vdots \\ \mathbf{H}_{ij_i} \end{pmatrix} \qquad (7)$$

$$\mathbf{G}_{\mathcal{H}_i} = \left.\frac{\partial \mathbf{f}_{\mathcal{H}_i}}{\partial \mathbf{y}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}})} = \begin{pmatrix} \mathbf{G}_{\mathcal{H}_{i-1}} \\ \mathbf{G}_{ij_i} \end{pmatrix} = \begin{pmatrix} \mathbf{G}_{1j_1} \\ \vdots \\ \mathbf{G}_{ij_i} \end{pmatrix} \qquad (8)$$

The Joint Compatibility of pairings belonging to $\mathcal{H}_i$ can be determined using an innovation test on the joint innovation $\mathbf{h}_{\mathcal{H}_i}$ as follows:

$$\begin{aligned}
\mathbf{C}_{\mathcal{H}_i} &= \mathbf{H}_{\mathcal{H}_i}\mathbf{P}\mathbf{H}_{\mathcal{H}_i}^T + \mathbf{G}_{\mathcal{H}_i}\mathbf{S}\mathbf{G}_{\mathcal{H}_i}^T \qquad (9) \\
D_{\mathcal{H}_i}^2 &= \mathbf{h}_{\mathcal{H}_i}^T \mathbf{C}_{\mathcal{H}_i}^{-1} \mathbf{h}_{\mathcal{H}_i} < \chi_{d,\alpha}^2 \qquad (10)
\end{aligned}$$

where $\mathbf{C}_{\mathcal{H}_i}$ is the covariance of the joint innovation, α is the desired confidence level, and $d = \dim(\mathbf{f}_{\mathcal{H}_i})$. The size of both $\mathbf{h}_{\mathcal{H}_i}$ and $\mathbf{C}_{\mathcal{H}_i}$ increase with the size of hypothesis $\mathcal{H}_i$. This makes this test potentially expensive to apply. The calculation of lower bounds of $D_{\mathcal{H}_i}^2$ to try to determine whether the hypothesis can be rejected without inverting $\mathbf{C}_{\mathcal{H}_i}$ is pointless in this case, since all the pairings in $\mathcal{H}_i$ are already known to be individually compatible.

In the worst case, verifying the joint compatibility of a set of pairings in a hypothesis involves the inversion of an $m \times m$ matrix, $O(m^3)$. This complexity can be reduced to $O(m^2)$ using the fact that joint compatibility can be *incrementally* evaluated: given a hypothesis $\mathcal{H}_{i-1}$ formed by jointly compatible pairings , with its corresponding $\mathbf{h}_{\mathcal{H}_{i-1}}$, $\mathbf{C}_{\mathcal{H}_{i-1}}^{-1}$, and $D_{\mathcal{H}_{i-1}}^2$, and given a pairing (E_i, F_{j_i}), with corresponding $\mathbf{h}_{ij_i}$, and $\mathbf{C}_{ij_i}$, the Mahalanobis distance $D_{\mathcal{H}_i}^2$ corresponding to $\mathcal{H}_i$ can be calculated as follows:

1) Calculate $\mathbf{C}_{\mathcal{H}_i}^{-1}$. From eq. (9) we have that:

$$\begin{aligned}
\mathbf{C}_{\mathcal{H}_i} &= \begin{pmatrix} \mathbf{H}_{\mathcal{H}_{i-1}} \\ \mathbf{H}_{ij_i} \end{pmatrix} \mathbf{P} \begin{pmatrix} \mathbf{H}_{\mathcal{H}_{i-1}}^T & \mathbf{H}_{ij_i}^T \end{pmatrix} \\
&\quad + \begin{pmatrix} \mathbf{G}_{\mathcal{H}_{i-1}} \\ \mathbf{G}_{ij_i} \end{pmatrix} \mathbf{S} \begin{pmatrix} \mathbf{G}_{\mathcal{H}_{i-1}}^T & \mathbf{G}_{ij_i}^T \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{C}_{\mathcal{H}_{i-1}} & \mathbf{w}_i^T \\ \mathbf{w}_i & \mathbf{C}_{ij_i} \end{pmatrix}
\end{aligned}$$

where:

$$\mathbf{w}_i = \mathbf{H}_{ij_i}\mathbf{P}\mathbf{H}_{\mathcal{H}_{i-1}}^T + \mathbf{G}_{ij_i}\mathbf{S}\mathbf{G}_{\mathcal{H}_{i-1}}^T$$

According to the partitioning method for matrix inversion [24], matrix $\mathbf{C}_{\mathcal{H}_i}^{-1}$ can be calculated as:

$$\mathbf{C}_{\mathcal{H}_i}^{-1} = \begin{pmatrix} \mathbf{K}_i & \mathbf{L}_i^T \\ \mathbf{L}_i & \mathbf{N}_i \end{pmatrix} \qquad (11)$$

where:

$$\begin{aligned}
\mathbf{N}_i &= \left(\mathbf{C}_{ij_i} - \mathbf{w}_i \mathbf{C}_{\mathcal{H}_{i-1}}^{-1} \mathbf{w}_i^T\right)^{-1} \\
\mathbf{L}_i &= -\mathbf{N}_i \mathbf{w}_i \mathbf{C}_{\mathcal{H}_{i-1}}^{-1} \\
\mathbf{K}_i &= \mathbf{C}_{\mathcal{H}_{i-1}}^{-1} + \mathbf{L}_i^T \mathbf{N}_i^{-1} \mathbf{L}_i
\end{aligned}$$

2) Calculate $D_{\mathcal{H}_i}^2$:

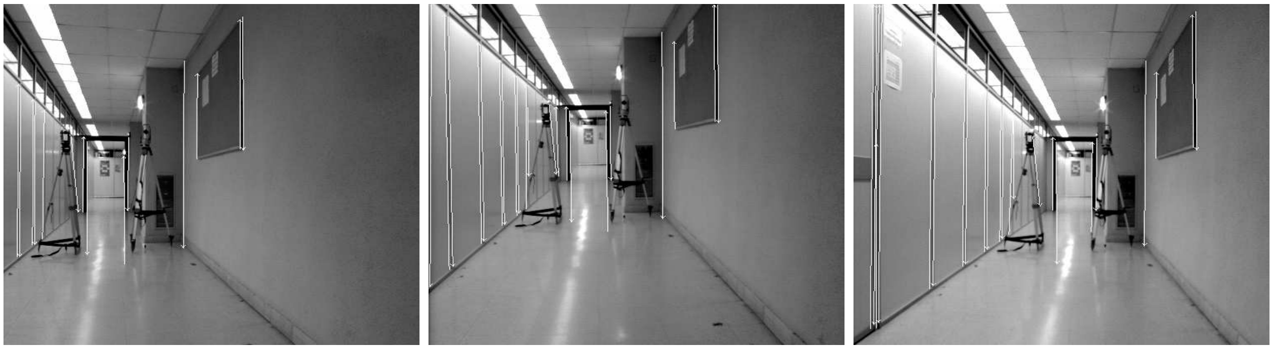


Fig. 3. Images of the trinocular vision system at step 48 of the robot trajectory.

$$\begin{aligned}
D^2_{\mathcal{H}_i} &= \mathbf{h}^T_{\mathcal{H}_i}\mathbf{C}^{-1}_{\mathcal{H}_i}\mathbf{h}_{\mathcal{H}_i} \\
&= \begin{pmatrix} \mathbf{h}^T_{\mathcal{H}_{i-1}} & \mathbf{h}^T_{ij_i} \end{pmatrix} \begin{pmatrix} \mathbf{K}_i & \mathbf{L}_i^T \\ \mathbf{L}_i & \mathbf{N}_i \end{pmatrix} \begin{pmatrix} \mathbf{h}_{\mathcal{H}_{i-1}} \\ \\ \mathbf{h}_{ij_i} \end{pmatrix} \\
&= \mathbf{h}^T_{\mathcal{H}_{i-1}}\mathbf{K}_i\mathbf{h}_{\mathcal{H}_{i-1}} + 2\mathbf{h}^T_{ij_i}\mathbf{L}_i\mathbf{h}_{\mathcal{H}_{i-1}} + \mathbf{h}^T_{ij_i}\mathbf{N}_i\mathbf{h}_{ij_i} \\
&= D^2_{\mathcal{H}_{i-1}} + \mathbf{h}^T_{\mathcal{H}_{i-1}}\mathbf{L}_i^T\mathbf{N}_i^{-1}\mathbf{L}_i\mathbf{h}_{\mathcal{H}_{i-1}} \\
&\quad + 2\mathbf{h}^T_{ij_i}\mathbf{L}_i\mathbf{h}_{\mathcal{H}_{i-1}} + \mathbf{h}^T_{ij_i}\mathbf{N}_i\mathbf{h}_{ij_i} \qquad (12)
\end{aligned}$$

An alternative method for the incremental calculation of the Mahalanobis distance can be found in [21].

Joint Compatibility is equivalent to Sequential Compatibility (except for linealization errors): if $D^2_{\mathcal{H}_i}$ satisfies eq. (10), the pairing (E_i, F_{j_i}) is compatible with all pairings in $\mathcal{H}_{i-1}$. There are several advantages in using JC:

- JC can be tested without recomputing the state $\hat{\mathbf{x}}$, an $O(n^2)$ operation.
- The inversion of a growing $\mathbf{C}_{\mathcal{H}_i}$ matrix is avoided. In fact, only the inversion of a constant size matrix $\mathbf{N}_i$ is necessary.
- Most of the elements involved in the calculation of $D^2_{\mathcal{H}_i}$ ($\mathbf{h}_{ij_i}$, $\mathbf{H}_{ij_i}$, $\mathbf{C}_{ij_i}$) have been previously computed to assert Individual Compatibility.

The Joint Compatibility Branch and Bound (JCBB) algorithm proposed in this work traverses the interpretation tree in search for the hypothesis with the largest number of non-null jointly compatible pairings. This monotonically non-decreasing criterion can be used to *bound* the search in the interpretation tree [2]. The quality of a node at level i, corresponding to a hypothesis $\mathcal{H}_i$, can be defined as the number of non-null pairings that can be established from the node. In this way, nodes with quality lower than the best available hypothesis are not explored. The nearest neighbor rule using the Mahalanobis distance $D^2_{\mathcal{H}_i}$ can be used as heuristic for *branching*, so that the nodes corresponding to hypotheses with a higher degree of joint compatibility will be explored first.

As experiments will show, Joint Compatibility is a very restrictive criterion to traverse the interpretation tree, limiting the combinatorial explosion of the search due to increasing vehicle error.

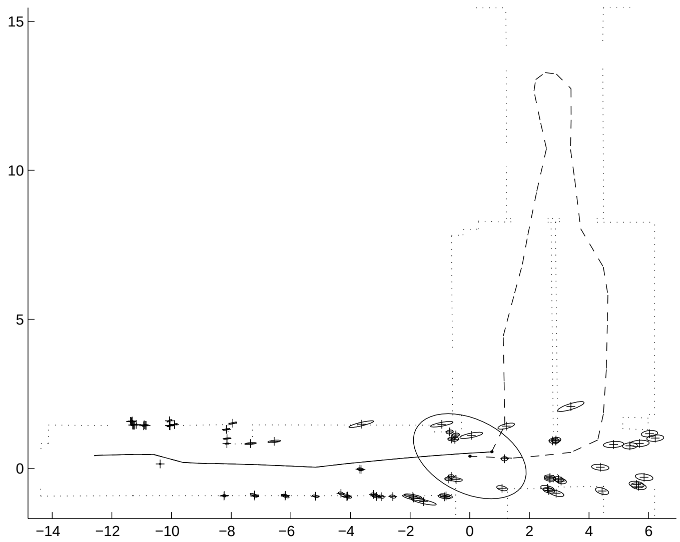


Fig. 4. Stochastic map obtained during the first 18 steps of the robot trajectory (solid line) and true robot trajectory until step 48 (dashed line), with resulting uncertainty in the robot location after continuing the robot trajectory without updating the stochastic map. A map of the environment (dotted line) is superimposed as reference but not used during the process.

IV. Experiments

A Labmate mobile robot equipped with a trinocular vision system was programmed to perform a loop trajectory of around $60m$. The ground truth for the robot location along the trajectory was obtained using theodolites. Trinocular vision provided a set of 2D points corresponding to corners, wall and window frames, etc., obtained by establishing correspondences between vertical edges extracted from the three images (fig. 3).

Continuous localization and map building using this information is fairly straightforward. Thus, we designed an experiment to determine how well the SCNN and JCBB data association algorithms solve the more complex and important *revisiting problem*. A stochastic map of a corridor was constructed [5] using the information obtained in the first 18 steps of the robot trajectory (fig. 4). The robot trajectory was continued *without* updating the stochastic map until the first corridor was visible again (step 48). Fig. 5 depicts the situation from the robot's perspective: the measurements obtained by the trinocular vision system, and the predicted location of map

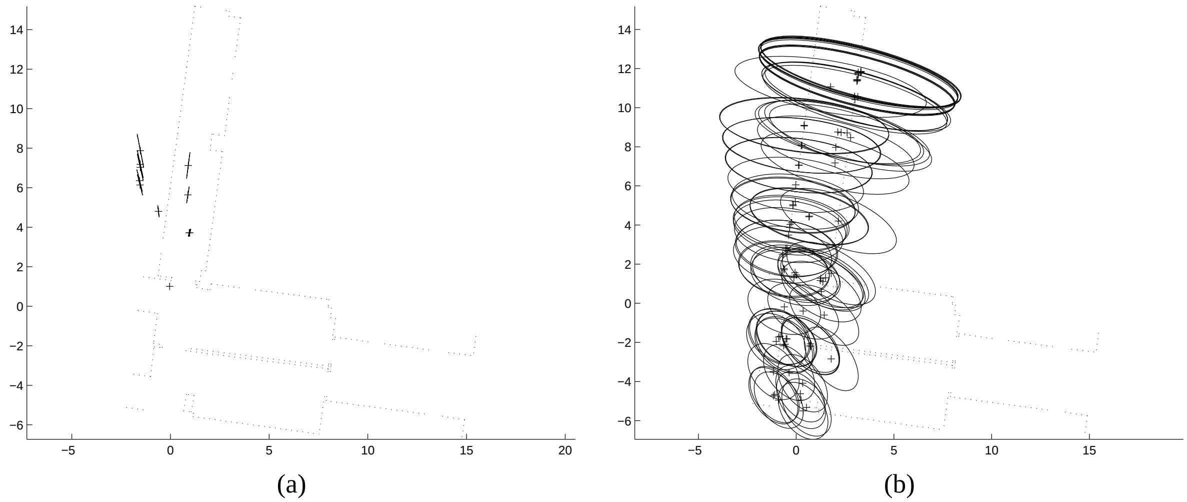


Fig. 5. The revisiting problem at trajectory step 48: (a) 2D points obtained by trinocular vision and (b) predicted location of map features with respect to the robot location.

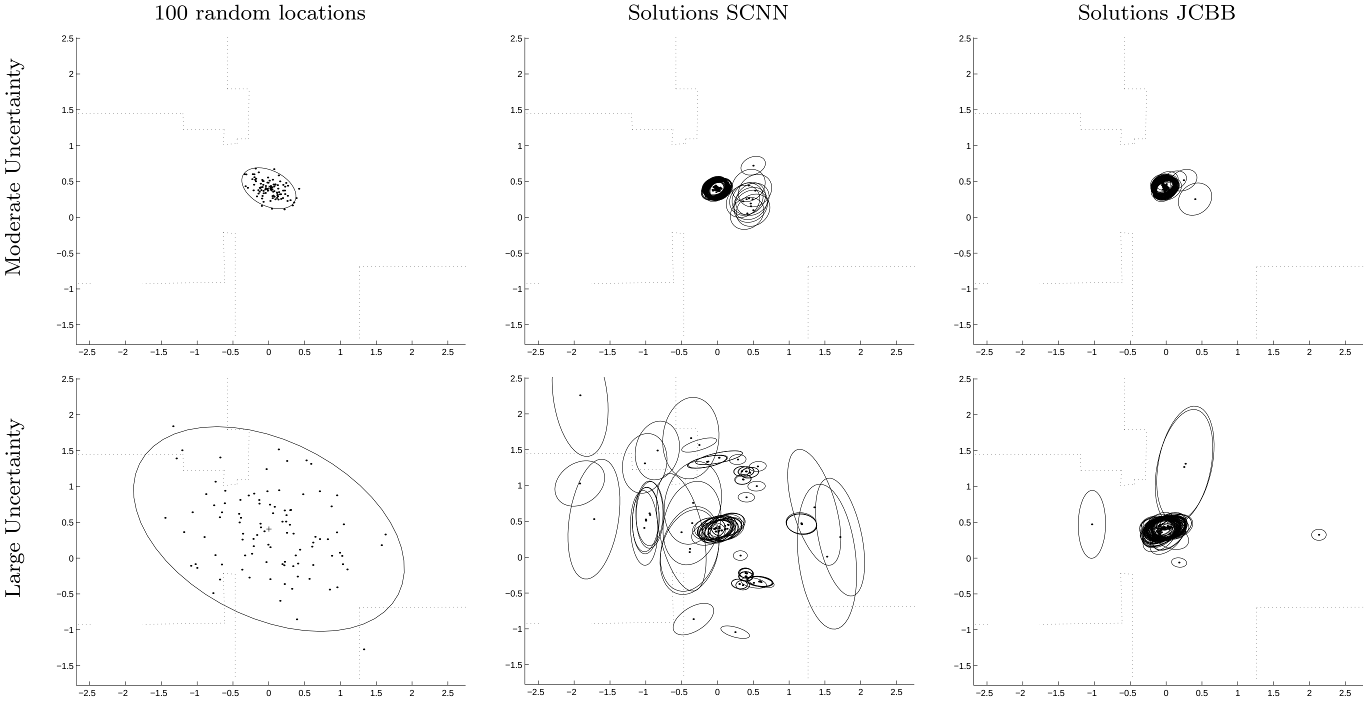


Fig. 6. Solutions for 100 random robot locations using SCNN and JCBB for moderate and for large uncertainty.

features relative to the robot. It is clear that robot error and environment clutter make the number of candidate features for each measurement large and the situation very confusing.

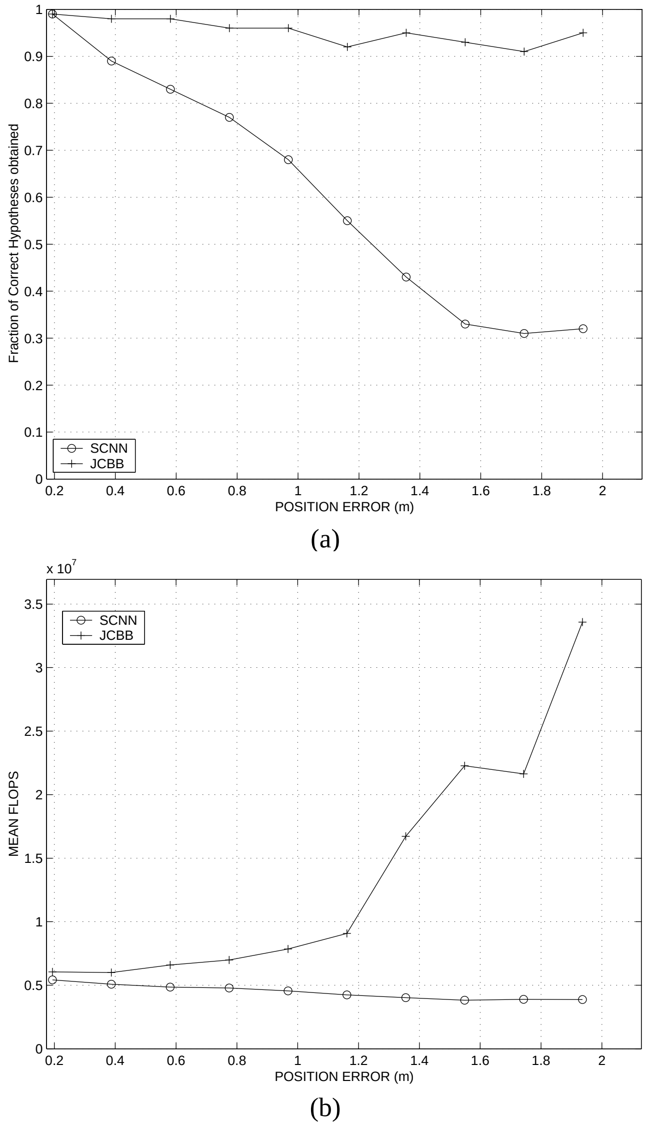
A Monte Carlo simulation was performed to generate 100 random positions at 10 different fractions of the odometry uncertainty at step 48. Both SCNN and JCBB were then executed, and the estimations of the robot location according to the resulting hypotheses were compared with ground truth. In the following, we discuss the results from two perspectives: robustness to robot error and computational efficiency.

A. Robustness of SCNN .vs. JCBB

Fig. 6 shows results for frontal, lateral and angular (2σ) errors $0.15m$, $0.11m$, and $1.4deg$ (top) and $1.55m$, $1.16m$, and $14deg$ (bottom). It can be seen that SCNN exhibits a great dispersion in the solutions obtained, more notable as robot error grows. In contrast, the solutions obtained by JCBB remain grouped around the true robot location even for large uncertainty values.

In order to measure the robustness of both algorithms to the increase in robot error, the 100 hypotheses obtained by each were compared with the true associations, established by hand. Results (fig. 7.a) show that the robustness of SCNN drops quickly with the increase in error, making this algorithm unsuitable for loop closing situations. In contrast, the probability of success of JCBB is always higher than that of SCNN, remaining above 0.9.

It should be noted that the Joint Compatibility test is based



(a)

(b)

Fig. 7. Performance of JCBB and SCNN .vs. robot position 2σ error: (a) fraction of correct hypotheses (with no spurious pairings) obtained and (b) computational cost (in FLOPS).

on the linearization of the relation between the measurements and the state (eq. 2). JCBB will remain robust to robot error as long the linear approximation is reasonable. Thus, the adequacy of using JCBB is determined by the robot orientation error (in practice, we have found the limit to be around $30deg$).

B. Computational Efficiency of SCNN .vs. JCBB

That JCBB is more robust to an increment of robot error than SCNN was an expected result: JCBB is a back-tracking algorithm, while SCNN is a greedy search algorithm. Given that the cost of SCNN is linear with the number of measurements m, the important question is then: is JCBB, an exponential time algorithm, feasible? This essentially depends on two factors: (1) how restrictive JC is to limit the real size of the solution space, and (2) how effective the branch and bound is to limit the search for the best solution within this space.

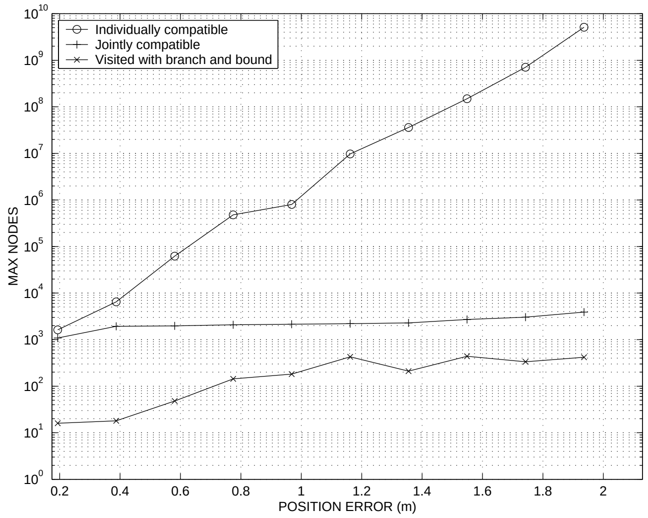


Fig. 8. Maximum number of nodes .vs. position error (2σ).

Fig. 8 shows the variation in the maximum number of individually compatible nodes, jointly compatible nodes, and visited nodes, all with respect to the increase in error. Joint Compatibility shows to be a highly restrictive criterion. We can see that the total number of nodes in the solution space grows several orders of magnitude between $0.2m$ and $2m$ in position error. But this is not the case for the maximum number of jointly compatible nodes and maximum number of nodes visited by branch and bound. This is because the joint compatibility of a certain number of 2D points fundamentally depends on their *relative error* (which depends on sensor and map precision), more than on their *absolute error* (which depends on robot error).

Results (fig. 7.b) show that the computational cost of SCNN decreases slightly with the increase in robot error, due to the fact that a more probable spurious pairing normally reduces the possibility of establishing more pairings. For small robot position error, the cost of JCBB and SCNN are similar. For a moderate error, around $1m$ in position and $7deg$ in orientation, JCBB executes around twice more flops. For larger errors, up to $2m$ and $14deg$, the computational cost of JCBB increases rapidly, although it remains feasible in real time. It is important to note that these results refer to edge-based trinocular vision, an especially difficult data association problem. In simpler cases, such as walls observed with a laser rangefinder, the method will handle larger errors, up to the limit imposed by the linearization errors.

Fig. 9 shows the computational cost of JCBB with respect to the increase in the number of observations m. It can be seen that the cost of establishing individual compatibility and the cost of updating the stochastic map grow linearly. On the other hand, it is a known fact that the cost of the interpretation tree search in the presence of spurious observations grows exponentially [2]. Our experiments show an asymptotic complexity of $O(1.53^m)$. This means that the number of observations must be limited to make this algorithm feasible in real time. In practice, selecting the 10 or 12 more relevant observations (larger or more precise) guarantees the robustness of the

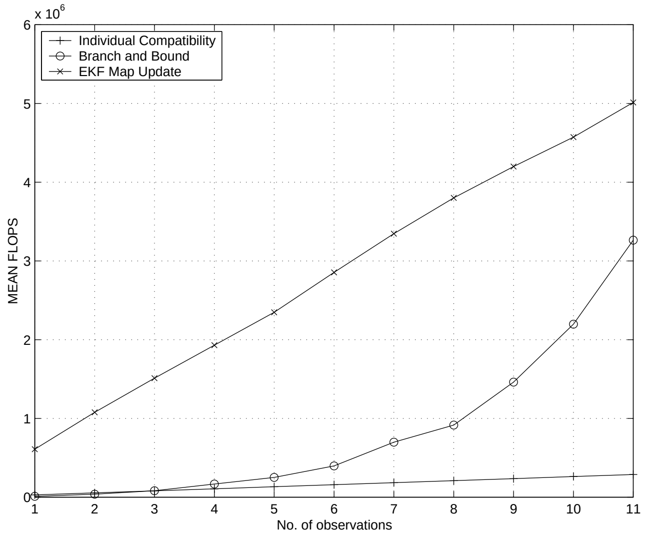


Fig. 9. Computational cost of the different parts of our JCBB algorithm .vs. number of observations.

hypothesis obtained by the JCBB algorithm. Once the map is updated using this hypothesis, the remaining observations can be safely associated with the SCNN algorithm.

V. Conclusions

The popular Nearest Neighbor algorithm for data association in stochastic mapping is very sensitive to the increase in vehicle and sensor error. Both factors contribute to increase the probability of matching a sensor measurement with an unrelated map feature. Since NN does not reconsider the establishment of a measurement-feature pairing, spurious pairings are easily formed and never reconsidered.

We have shown that reconsideration of the validity of pairings is necessary, and thus a constrained search algorithm, as the JCBB algorithm described in this paper, is required. The combinatorial explosion of this backtracking algorithm with respect to increase in vehicle error is controlled by the use of a restrictive validation mechanism that determines the joint compatibility of a set of pairings in the hypothesis. This greatly reduces the number of nodes in the interpretation tree that must be visited, as the algorithm traverses the tree in higher levels. Experiments show that JCBB is not only more robust than NN algorithms, but also feasible in terms of computational cost.

In this work we have studied the problem of refining an available estimation of the vehicle and feature state. Complementary techniques, applicable when there is no estimation of the vehicle location, will constitute further work.

References

[1] T. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press In., Boston, Mass., 1988.

[2] W.E.L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, The MIT Press, Cambridge, Mass., 1990.

[3] R. Smith, M. Self, and P. Cheeseman, “A stochastic map for uncertain spatial relationships,” in *4th Int. Symp. Robotics Research*, O. Faugeras and G. Giralt, Eds., pp. 467–474. The MIT Press, 1988.

[4] Z. Zhang and O.D. Faugeras, “A 3D world model builder with a mobile robot,” *Int. J. Robotics Research*, vol. 11, no. 4, pp. 269–285, 1992.

[5] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós, “The SPmap: A probabilistic framework for simultaneous localization and map building,” *IEEE Trans. Robot . Automat.*, vol. 15, no. 5, pp. 948–953, 1999.

[6] H.J.S. Feder, J.J. Leonard, and C.M. Smith, “Adaptive mobile robot navigation and mapping,” *Int. J. Robotics Research*, vol. 18, no. 7, pp. 650–668, 1999.

[7] P.M. Newman, *On The Structure and Solution of the Simultaneous Localisation and Map Building Problem*, Ph.D. thesis, Dept. Mechanical and Mechatronic Engineering, University of Sydney, March 1999.

[8] J.J. Leonard and R.J. Rikoski, “Incorporation of delayed decision making into stochastic mapping,” in *Experimental Robotics VII, Lecture Notes in Control and Information Sciences*, D. Rus and S. Singh, Eds. Springer Verlag, 2001.

[9] J.A. Pérez, J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós, “Continuous mobile robot localization: Vision .vs. laser,” in *IEEE Int. Conf. Robotics and Automation*, Detroit, USA, 1999, pp. 2917–2923.

[10] I. J. Cox and J. J. Leonard, “Modeling a dynamic environment using a multiple hypothesis approach,” *A.I. Journal*, vol. 66, no. 2, pp. 311–344, 1994.

[11] C.M. Smith, *Integrating Mapping and Navigation*, Ph.D. thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, MA, June 1998.

[12] X. Lebegue and J.K. Aggarwal, “Generation of architectural CAD models using a mobile robot,” in *IEEE Int. Conf. Robotics and Automation*, San Diego, California, 1994, pp. 711–717.

[13] C.F. Olson, “Probabilistic self-localization for mobile robots,” *IEEE Trans. Robot. Automat.*, vol. 16, no. 1, pp. 55–66, 2000.

[14] S. Thrun, W. Burgand, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping,” in *IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 2000, pp. 321–328.

[15] J.S. Gutmann and K. Konolige, “Incremental mapping of large cyclic environments,” in *IEEE Int. Symp. on Computational Intelligence in Robotics and Automation CIRA’1999*, 1999, pp. 318–325.

[16] T. Baley, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte, “Data association for mobile robot navigation: A graph theoretic approach,” in *IEEE Int. Conf. Robotics and Automation*, San Francisco, California, 2000, pp. 2512–2517.

[17] J.A. Castellanos and J.D. Tardós, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*, Kluwer Academic Publishers, Boston, Mass., 1999.

[18] J. Neira, J.D. Tardós, J. Horn, and G. Schmidt, “Fusing range and intensity images for mobile robot localization,” *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 76–84, 1999.

[19] J.B. Collins and J.K Uhlmann, “Efficient gating in data association with multivariate gaussian distributed states,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 909–916, July 1992.

[20] M.J.L. Orr, J. Hallam, and R.B Fisher, “Fusion through interpretation,” in *Second European Conf. Computer Vision*, 1992.

[21] J.M.M Montiel and L. Montano, “Efficient validation of matching hypotheses using mahalanobis distance,” *Engineering Applications of Artificial Ingelligence*, vol. 11, no. 3, pp. 439–448, 1998.

[22] J.J. Leonard and H.J.S. Feder, “A computationally efficient method for large-scale concurrent mapping and localization,” in *9th Int. Symp. Robotics Research*, D. Koditschek and J. Hollebach, Eds., pp. 169–176. Springer Verlag, Snowbird, Utah, 2001.

[23] J. E. Guivant and E. M. Nebot, “Optimization of the simultaneous localization and map building algorithm for real-time implementation,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 242–257, 2001.

[24] David A. Harville, *Matrix Algebra from a Statistician’s Perspective*, Springer-Verlag, New York, Inc., 1997.