Multi-Vehicle Cooperative Local Mapping: A Methodology Based on Occupancy Grid Map Merging

Hao Li, Manabu Tsukada, Fawzi Nashashibi, Michel Parent

Multi-Vehicle Cooperative Local Mapping: A Methodology Based on Occupancy Grid Map Merging

Hao LI, Manabu TSUKADA, Fawzi NASHASHIBI, Michel PARENT

Abstract—Local mapping is valuable for many real-time applications of intelligent vehicle systems. Multi-vehicle *cooperative local mapping* can bring considerable benefits for vehicles operating in some challenging scenarios. In this paper, we introduce a method of occupancy grid map merging, dedicated to multi-vehicle cooperative local mapping purpose in outdoor environment. In a general map merging framework, we propose an objective function based on occupancy likelihood, and provide some concrete procedures designed in the spirit of genetic algorithm to optimize the defined objective function. Based on the introduced method, we further describe a strategy of indirect vehicle-to-vehicle relative pose estimation, which can serve as a general solution for multi-vehicle perception association. We present a variety of experiments that validate the effectiveness of the proposed occupancy grid map merging method. We also demonstrate several useful application examples of the indirect vehicle-to-vehicle relative pose estimation strategy.

Index Terms—Cooperative perception, cooperative local mapping, occupancy grid map, map merging, relative pose estimation, intelligent vehicle systems

I. INTRODUCTION

Since several decades ago, researchers have been making efforts on realizing various vehicle-related intelligent functions, such as vehicle localization [1] [2], object detection [3] [4], vehicle control [5] [6], etc. Among these functions, autonomous *mapping*, which is often in the form of SLAM (simultaneous localization and mapping) [7] [8] [9], has long since been a fundamental one for vehicles (or mobile robots) operating in unknown environment. *Local mapping* is especially valuable for many real-time applications in outdoor environment [10] [11]—here, for a vehicle, *mapping* implies its perception towards its general surrounding entities (not limited to fixed objects).

A vehicle can perform local mapping based only on its own data. In contrast, multi-vehicle *cooperative local mapping*, which employs vehicular communication standard technology [12] [13] besides traditional on-vehicle sensors, can bring considerable benefits for vehicles operating in some challenging scenarios. Take the overtaking scenario as an example, see Fig.1. This typical scenario is potentially dangerous, because the first vehicle (overtaken vehicle) occludes the view of the second vehicle (overtaking vehicle). For safety and efficiency reason, the second vehicle wonders: *what are there occluded by the first vehicle*? Unfortunately, it can never answer this question by itself, simply because it cannot perceive the occluded environment.

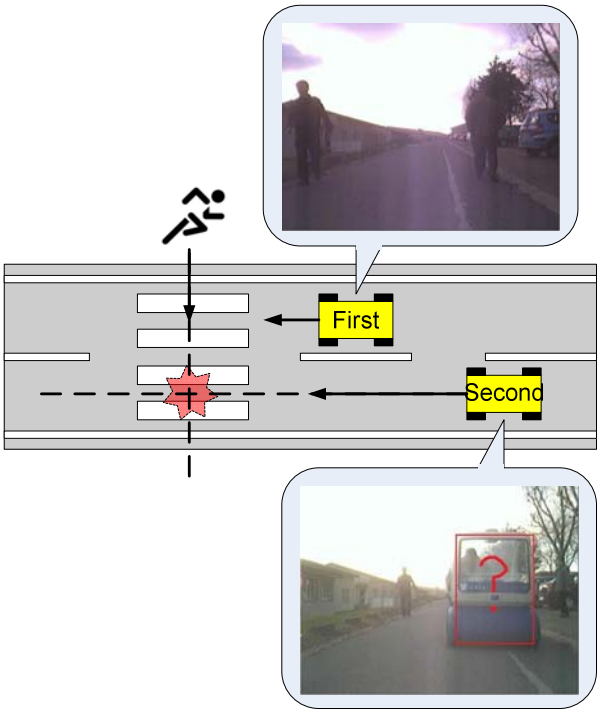

Fig.1. Overtaking scenario: potentially dangerous because of view occlusion

Cooperative local mapping can help the second vehicle answering this question: the second vehicle can import the local map built by the first vehicle and indirectly "perceive" the occluded environment. For realizing this, a fundamental requirement consists in correct *local map merging*, i.e. consistently aligning the two local maps.

In this paper, we introduce a method of occupancy grid map merging, dedicated to multi-vehicle cooperative local mapping purpose in outdoor environment. In section II, we give a state-of-the-art on map representation and map merging; then we explain why to adopt the occupancy grid representation and to propose our occupancy grid map merging method. In section III, we review some techniques for a vehicle to perform local SLAM and global localization. In section IV, we detail our occupancy grid map merging method, based on which we further describe in section V a strategy of indirect vehicle-to-

vehicle relative pose estimation; this strategy can serve as a general solution for multi-vehicle perception association. In section VI, we present a variety of experiments on the method of occupancy grid map merging, and briefly demonstrate several applications of the indirect vehicle-to-vehicle relative pose estimation strategy, followed by the conclusion section.

II. Related Works

A. Map Representation

There exist several typical representations for a map (we deal with 2-D representation): direct representation, feature-based representation, and occupancy grid representation.

Direct representation: a map can be represented as a registered list of raw sensor scans [14] [15]. The direct representation exempts a perception system from sensor data conversion; however, it has some disadvantages: it neither models the map uncertainties nor models the environment part that a sensor cannot directly measure, such as free space. In addition, managing a map in such representation is not easy; for example, how to handle the overlapping part of raw scans? How to organize the data so that we can efficiently access and use them? These issues are not trivial.

Feature-based representation: a map can be represented as a group of features extracted from sensor data. The features can be artificial landmarks [16]; they can be natural landmarks such as outdoor trees [17] or indoor line segments [18]; they can be of abstract form, detected by certain specific algorithms [19] [20]. This representation enjoys low memory requirement and the flexibility to adjust the map elements. However, its apparent disadvantage is its inability to represent general unstructured environment, as it requires extracting certain features and represents only these features.

Occupancy grid representation: the occupancy grid [21] is a 2-D lattice that divides the ground plane into rectangular cells. A cell is associated with a real value in [0,1], which indicates the state of the cell being occupied by or free of object. Cell value 0.5 means the cell being in unknown state. The larger the cell value is, the more the cell tends to be occupied; the smaller the cell value is, the more the cell tends to be free. The occupancy grid representation has some merits: above all, it can represent *general unstructured* environment. Besides, it can model the map uncertainty. Moreover, an occupancy grid map is similar to our daily maps, which makes it suitable to our normal habit for dealing with maps.

We adopt the *occupancy grid* representation for its merits mentioned. As a dense representation, it might suffer from forbidding memory burden, if we aim at global mapping for a large area. However, for many real-time vehicle applications, global mapping is unnecessary, whereas *local mapping*, which enables a vehicle to monitor its surroundings in real time, is more valuable [10] [11]. For local mapping, the occupancy grid representation can be easily controlled at tractable size.

B. Map Merging

On the whole, there are two categories of map merging methods: vehicle pose estimation based methods, and map consistency based methods.

1) Vehicle pose estimation based methods

If two vehicles are precisely localized in a global reference, their local maps can be mutually related using their precise global localization result. However, precise global localization of a vehicle cannot be always expected: a vehicle with normal GPS often has a positioning error as large as several meters. A vehicle even with RTK-GPS might still suffer from signal degradation. Thus, map merging based on vehicle global localization result is generally impractical.

A practice for map merging is based on direct vehicle-to-vehicle relative pose estimation, where a vehicle directly detects another vehicle and estimates their relative pose from the detection result [22] [23] [24] [25]. Direct vehicle-to-vehicle relative pose estimation requires performing three sub-processes: detection, association, and relative pose estimation; they are not easy to handle in outdoor environment which is dynamic and full of certain occlusions. Both vehicle detection and detected result association are challenging problems that deserve continuous research. Even if the detection and the association are well performed, it is still difficult to extract the geometry of a vehicle from a detection result which always corresponds to partial contour (sometimes even irregular) of the vehicle. A*d hoc* patterns (with special colors, shapes etc) can facilitate the three sub-processes [23] [26] [27]. However, designing proper patterns to distinguish thousands of outdoor vehicles is not easy; moreover, occlusions might cause miss detection and false detection of these patterns.

2) Map consistency based methods

These methods do not require accurate estimation of vehicle poses (global or relative); they try to find the map alignment directly, based on certain consistency measure between the maps. Various map consistency based methods exist, reported in literature from different views. For example, if we treat a map as a scan of data, map merging becomes *scan matching* [28]. To avoid confusion, we neglect different terminology and here treat related methods uniformly as *map merging*. We can categorize these methods using different criteria. For example, concerning map representation, some methods deal with the direct one [28], some with the feature-based one [18], and some with the occupancy grid one [29]. If we examine the merging strategy, we can roughly assign these methods to three categories: feature-based methods, iterative closest point (ICP) methods, and overall-direct optimization methods.

Feature-based methods: In a feature-based method, certain features are used to facilitate the merging [18] [30] [31]—It is worthy noting that A feature-based method is not necessarily juxtaposed with the feature-based representation. For example, the merging of two occupancy grid maps can be based on only some features extracted from the maps [32] [33]—Similar to feature-based representation, feature-based methods are usually limited to the environment in which certain features are reliably extractable.

ICP methods: the ICP strategy is a popular architecture for merging general maps, without feature extraction. It originated from [34] [35], with a family of following variants [36]. It consists in an iteration of two steps: 1) tentatively establish a new set of correspondences, heuristically based on the closest point rule; 2) update the merging result by minimizing a distance measure defined by the new set of correspondences. Given a suitable initial value, the merging result normally converges after a number of iterations of the two steps. The metric originally used in the ICP is the Euclidean distance.

Later, more general metrics [28] [37] have been incorporated in the ICP to establish correspondences, which might better capture the map alignment.

Overall-direct optimization methods: an overall-direct optimization method generally consists of two parts: 1) define an objective function of the map alignment to characterize the overall consistency degree between two maps; 2) optimize this objective function. For example, an objective function based on a normal distribution transform can be used as the consistency measure, and optimized by the Newton's algorithm [38]. For occupancy grid map merging, an objective function with a *similarity* term and a *lock* term can be used, and optimized by the random walk algorithm [29].

Overall-direct optimization methods and ICP methods both deal with the *overall* consistency between two maps. For an ICP method, as a new set of correspondences is established, the optimization problem changes and has to be solved again. For an overall-direct optimization method, the optimization problem is fixed and needs to be solved only once; in other words, its optimization architecture is comparatively *direct*. This explains the indication of *overall* and *direct* for overall-direct optimization methods.

In outdoor environment, an initial map alignment can be estimated using GPS based vehicle global localization result, yet it is often not close enough to the correct one. Thus, we do not rely on ICP methods that require suitable initial values. We do not rely either on feature-based methods which are not generally applicable. Instead, we prefer to follow an overall-direct optimization method.

C. Occupancy Grid Map Merging

We adopt the occupancy grid representation and follow the overall-direct optimization spirit to merge two occupancy grid maps, as in [29]. The objective function given in [29] consists of a *similarity* term, which represents the overall distance between the maps, and a *lock* term heuristically added, which counteracts the over-fitting effect. This objective function has two disadvantages: first, it is susceptible to map inherent inconsistency, i.e. the map inconsistency that exists even if the maps are aligned correctly—for example, map inherent inconsistency can be caused by the view difference between two vehicles—Map inherent inconsistency can cause errors in the distance term and false counting of *disagreement* in the lock term, which can mislead the merging result. Second, the parameter in the lock term has to be tuned empirically, which is an undesirable point concerning the method generality.

We define an objective function that neither is susceptible to map inherent inconsistency nor requires tuning any heuristic parameter as in [29]. We design some concrete procedures in the spirit of genetic algorithm to efficiently optimize the defined objective function. The proposed method of occupancy grid map merging is detailed in section IV.

III. Vehicle Local SLAM and Global Localization

GPS, laser scanners, and motion sensors are often used in intelligent vehicles. For a vehicle with a normal GPS, a 2D laser scanner, and motion sensors, we briefly review some techniques on vehicle local SLAM and global localization.

A. Occupancy Grid Based Local SLAM

We adopt the incremental maximum likelihood SLAM method [11], considering its computational efficiency, its local mapping accuracy, and its insensitiveness to dynamic entities. Let $\mathbf{S}=[x_s,y_s,\theta_s]^{\mathrm{T}}$ denote the *vehicle local state* (different from the *vehicle global state* $\mathbf{X}$ described latter). Let $\mathbf{M}$ denote the occupancy grid map. Let $\mathbf{u}$ denote the vehicle motion data. Let $\mathbf{z}$ denote the range data. Subscript t denotes the time index. This SLAM method is an iteration of step (1) and step (2):

$$\hat{\mathbf{S}}_t = \arg\max_{\mathbf{S}_t}\{p(\mathbf{z}_t \mid \mathbf{S}_t, \hat{\mathbf{M}}_{t-1})\, p(\mathbf{S}_t \mid \hat{\mathbf{S}}_{t-1}, \mathbf{u}_t)\} \tag{1}$$

$$\hat{\mathbf{M}}_t = \hat{\mathbf{M}}_{t-1} \cup \{\hat{\mathbf{S}}_t, \mathbf{z}_t\} \tag{2}$$

Given the vehicle state and the map at time t-1, step (1) is to search the optimal $\mathbf{S}_t$ which maximizes the marginal likelihood of the vehicle state and the range data at time t; the scan matching technique in [11] is used for searching the optimal $\mathbf{S}_t$. Once the optimal $\mathbf{S}_t$ is estimated, step (2) is to update the map by fusing the new data $\{\mathbf{S}_t,\mathbf{z}_t\}$ into the old map $\mathbf{M}_{t-1}$ [11].

B. GPS Based Global Localization

Let $\mathbf{X}=[x,y,\theta]^{\mathrm{T}}$ denote the *vehicle global state*; (x,y) and θ denote the vehicle position and orientation respectively in a global plan reference. A GPS directly provides error-bounded global position measurement; motion sensors can track the vehicle state when GPS data are temporarily unavailable. To estimate the vehicle state, we follow an Extended Kalman Filter (EKF) that fuses the GPS data and the motion data.

Global state evolution (prediction):

The motion of a vehicle can be modeled according to the kinematic bicycle model (denoted in compact form by G):

$$\mathbf{X}_t = G(\mathbf{X}_{t-1}, \mathbf{u}_t) \tag{3}$$

$$\mathbf{\Sigma}_{\mathbf{X}_t} = \mathbf{G}_{\mathbf{X}} \mathbf{\Sigma}_{\mathbf{X}_{t-1}} \mathbf{G}_{\mathbf{X}}{}^T + \mathbf{G}_{\mathbf{u}} \mathbf{\Sigma}_{\mathbf{u}} \mathbf{G}_{\mathbf{u}}{}^T \tag{4}$$

$\mathbf{\Sigma}_{\mathbf{X}}$ denotes the state covariance. $\mathbf{u}$ denotes the motion data (they can be refined with local SLAM result); $\mathbf{\Sigma}_{\mathbf{u}}$ denotes the motion data covariance. $\mathbf{G}_{\mathbf{X}}$ and $\mathbf{G}_{\mathbf{u}}$ denote the Jacobian matrices of function G with respect to $\mathbf{X}$ and $\mathbf{u}$ respectively.

Global state update:

Let the GPS measurement be denoted as $\mathbf{z}_G=(x_G,y_G)$. The observation model can be described as (at time t):

$$\mathbf{z}_{G,t} = \mathbf{H}_G \mathbf{X}_t$$

$\mathbf{H}_{\mathrm{G}}=[\mathbf{I}_{2\mathrm{x}2}\ \mathbf{0}_{2\mathrm{x}1}]$; the $\mathbf{z}_G$ error is assumed to follow the Gaussian distribution $N(\mathbf{0},\mathbf{\Sigma}_{\mathbf{G},t})$. The vehicle state is updated via (5):

$$\begin{aligned} \mathbf{K} &= \mathbf{\Sigma}_{\mathbf{X}_t} \mathbf{H}_G^T (\mathbf{H}_G \mathbf{\Sigma}_{\mathbf{X}_t} \mathbf{H}_G^T + \mathbf{\Sigma}_{G,t})^{-1} \\ \mathbf{X}_t &= \mathbf{X}_t + \mathbf{K}(\mathbf{z}_{G,t} - \mathbf{H}_G \mathbf{X}_t) \\ \mathbf{\Sigma}_{\mathbf{X}_t} &= (\mathbf{I} - \mathbf{K}\mathbf{H}_G)\mathbf{\Sigma}_{\mathbf{X}_t} \end{aligned} \tag{5}$$

It is worthy noting that there are two vehicle states, i.e. the *vehicle local state* $\mathbf{S}$ and the *vehicle global state* $\mathbf{X}$, as described. The reason for **not** mixing them is that mixing them can cause inconsistent local mapping: a GPS update can cause noticeable discontinuity between the vehicle state estimates before and after the GPS update, which can further result in

inconsistent local mapping—Local SLAM, above all, is to establish consistent inner relationship among the vehicle states and the surroundings at consecutive time sequence.

IV. Occupancy Grid Map Merging

Given two vehicles A and B, let $\mathbf{X}_{A(t)}$ and $\mathbf{X}_{B(t)}$ denote their global states. Let $\mathbf{S}_{A(t)}$ and $\mathbf{S}_{B(t)}$ denote their local states. Let $\mathbf{M}_{A(t)}$ and $\mathbf{M}_{B(t)}$ denote their occupancy grid maps. Without loss of generality, we neglect time index t in the following. In this section, we describe our method for merging $\mathbf{M}_A$ and $\mathbf{M}_B$.

A. Merging Framework

The occupancy grid map merging problem can be handled in a general optimization framework: first, design an objective function in terms of two generic occupancy grid maps $\mathbf{M}_1$ and $\mathbf{M}_2$, i.e. $F_c(\mathbf{M}_1,\mathbf{M}_2)$, which measures their consistency degree. Second, search the optimal relative pose $\mathbf{p}_{BA}$ that optimizes the consistency measure between $\mathbf{M}_A$ and $\mathbf{p}_{BA} \oplus \mathbf{M}_B$ (see the *compounding* operation in the Appendix), as in (6):

$$\hat{\mathbf{p}}_{BA} = \arg\max_{\mathbf{p}_{BA}} F_c(\mathbf{M}_A, \mathbf{p}_{BA} \oplus \mathbf{M}_B) \tag{6}$$

B. Occupancy Likelihood Based Objective Function

We propose an objective function based on occupancy likelihood. Let the occupied cells with local maximum occupancy state (called *local maximum occupied cells*) in $\mathbf{M}_A$ and $\mathbf{M}_B$ be denoted as a set of 2-D points $C(\mathbf{M}_A)=\{\mathbf{c}_{A(1)}, \mathbf{c}_{A(2)}, \ldots, \mathbf{c}_{A(na)}\}$ and another set $C(\mathbf{M}_B)=\{\mathbf{c}_{B(1)}, \mathbf{c}_{B(2)}, \ldots, \mathbf{c}_{B(nb)}\}$. Let the occupancy state of a position $\mathbf{p}$ in a map $\mathbf{M}$ be denoted as $\mathbf{M}(\mathbf{p})$. Then the objective function F_c is defined as in (7):

$$F_c(\mathbf{M}_A, \mathbf{p}_{BA} \oplus \mathbf{M}_B) = \sum_{\mathbf{c}\in C(\mathbf{M}_B)} F_A(\mathbf{p}_{BA}, \mathbf{c}) + \sum_{\mathbf{c}\in C(\mathbf{M}_A)} F_B(\mathbf{p}_{BA}, \mathbf{c}) \tag{7}$$

$$F_A(\mathbf{p},\mathbf{c}) = \begin{cases} \mathbf{M}_A(\mathbf{p}\oplus\mathbf{c}) & \text{if } \mathbf{p}\oplus\mathbf{c} \in Occ(\mathbf{M}_A) \\ 0 & \text{else} \end{cases}$$

$$F_B(\mathbf{p},\mathbf{c}) = \begin{cases} \mathbf{M}_B(inv(\mathbf{p})\oplus\mathbf{c}) & \text{if } inv(\mathbf{p})\oplus\mathbf{c} \in Occ(\mathbf{M}_B) \\ 0 & \text{else} \end{cases}$$

The *Occ* means the set of occupied cells (extracted by a threshold). The *Occ* threshold is not intended to judge whether a grid cell is truly occupied by an object; it is only used to select grid cells that tend to be closer to an object. So there is much flexibility in setting this threshold; we set it as 0.6.

Some explanations hovering around the idea of designing the F_c in (7): we examine the local maximum occupied cells of a map, as they most likely correspond to true object locations. For a local maximum occupied cell in a map, its occupancy state in the other map contributes to this F_c, if and only if it tends to be closer to an object in the other map. In other words, this F_c only deal with the consistent part of two maps and naturally is insensitive to map inherent inconsistency.

For local maps of enough size, stable and consistent objects (buildings, infrastructures etc) are usually dominant factors, which always contribute to successful local map merging—the correct map alignment normally achieves distinguished high value with the proposed objective function.

If a map appears consistent to another map, this other map tends to also appear consistent to it. Thus, a fair simplification of (7), i.e. (8) that only computes the consistency of one map relative to the other without computing the converse part, can achieve desirable performance as well:

$$F_c(\mathbf{M}_A, \mathbf{p}_{BA} \oplus \mathbf{M}_B) = \sum_{\mathbf{c}\in C(\mathbf{M}_B)} F_A(\mathbf{p}_{BA}, \mathbf{c}) \tag{8}$$

For the objective function in [29], its distance term might smooth its value space. An idea is to incorporate this distance term into (8) (or (7)) in a similar way as in [29], and we have (9)—the *md* denotes the metric in [29]:

$$\begin{aligned} F_c(\mathbf{M}_A, \mathbf{p}_{BA} \oplus \mathbf{M}_B) = & \sum_{\mathbf{c}\in C(\mathbf{M}_B)} F_A(\mathbf{p}_{BA}, \mathbf{c}) \\ & - \alpha \sum_{\mathbf{c}\in C(\mathbf{M}_B)} \min_{\mathbf{c}_A \in Occ(\mathbf{M}_A)} \{md(\mathbf{c}_A, \mathbf{p}_{BA}\oplus\mathbf{c})\} \end{aligned} \tag{9}$$

We prefer yet (8) instead of (9) for two reasons: first, we try to avoid tuning any heuristic parameter as α; second, use of (9) under some α can cause false merging, whereas use of (8) can always succeed in the merging task—see experiments.

C. Optimization Using Genetic Algorithm (GA)

An initial $\mathbf{p}_{BA}$ can be estimated with vehicle global state estimate, yet it is often far away from the optimal one. In addition, the value space of F_c is normally multimodal and irregular. Thus, local optimization techniques tend to fail.

The strategy of evolutionary *genetic algorithm* (GA) [39] is adopted to optimize F_c. A reason for adopting it is that it is not susceptible to the value space features of an objective function. Besides, it can be well realized in a dynamic (or recursive) scheme for real-time vehicle operations.

GA is rather a methodology instead of a list of concrete procedures. As an analogy to species evolution under natural selection, its spirit is to evaluate the fitness of a group of tentative solution individuals, vary them via biologically inspired operations such as *crossover* and *mutation*, and keep better individuals for further evolution. The concrete procedures of putting this spirit into practice are problem oriented and can be specially designed and modified. The concrete procedures in our implementation are as follows:

1. **Initialization**: randomly initialize a population of $\mathbf{p}_{BA}$:

(1-a) Compute an initial $\mathbf{p}_{BA}$ with vehicle state estimates:

$$\mathbf{p}_{BA(init)} = \mathbf{S}_A \oplus inv(\mathbf{X}_A) \oplus \mathbf{X}_B \oplus inv(\mathbf{S}_B)$$

(1-b) In an error range around $\mathbf{p}_{BA(init)}$, randomly initialize a population of $\mathbf{p}_{BA}$ i.e. $\{\mathbf{p}_{BA(k)} | k=1,2,\ldots,n\}$. We deliberately exaggerate this error range to be $\pm$30 m in position and $\pm 30^o$ in orientation, in order that the true $\mathbf{p}_{BA}$ is always in this range around $\mathbf{p}_{BA(init)}$. For the representation of a generic individual $\mathbf{p}_{BA(k)}$, we do not make *bit (binary) string encoding* [39] on $\mathbf{p}_{BA(k)}$; instead, we directly handle the real-value vector form of $\mathbf{p}_{BA(k)}$ for implementation efficiency.

2. **Evolution**: iteratively perform the following steps:

(2-a) Compute the likelihood (or *fitness value*) of each individual in the population, according to (8).

(2-b) Compute the mean likelihood of the population. For an individual, if its likelihood is above the mean likelihood, assign it to the *elite group*; otherwise, to the *inferior group*.

$$F_{c(\text{mean})} = \frac{1}{n}\sum_{k=1}^{n} F_c(\mathbf{M}_A, \mathbf{p}_{BA(k)} \oplus \mathbf{M}_B)$$

$$\{\mathbf{p}_{BA(\text{elite})}\} = \{\mathbf{p}_{BA(i)} \mid F_c(\mathbf{M}_A, \mathbf{p}_{BA(i)} \oplus \mathbf{M}_B) \geq F_{c(\text{mean})}\}$$

$$\{\mathbf{p}_{BA(\text{inferior})}\} = \{\mathbf{p}_{BA(j)} \mid F_c(\mathbf{M}_A, \mathbf{p}_{BA(j)} \oplus \mathbf{M}_B) < F_{c(\text{mean})}\}$$

Dividing the population according to their mean likelihood is a simple yet effective way to decide which individuals tend to survive and have influence on the next generation.

(2-c) Mutate the individuals in the elite group: for an elite individual, if its mutation has higher likelihood than its own, replace it with its mutation; otherwise, keep it unchanged.

For $\mathbf{p}_{BA(k)} \in \{\mathbf{p}_{BA(\text{elite})}\}$
$\mathbf{p}_{BA(k)}^* = MUTATE(\mathbf{p}_{BA(k)})$
If $F_c(\mathbf{p}_{BA(k)}^*) > F_c(\mathbf{p}_{BA(k)})$, then $\mathbf{p}_{BA(k)} = \mathbf{p}_{BA(k)}^*$
End for

The best individual is an exception; it gets more times (100 times) of mutation. If no mutation is better, just keep the best individual unchanged; otherwise, keep the best mutation.

(2-d) Replace the inferior group: for an inferior individual, replace it with a new individual that is generated by randomly applying the following genetic operations:

(2-d-i) Randomly select an individual from the elite group and mutate it to be the new individual—the best individual has to be selected at least once and mutated.

(2-d-ii) Randomly select two individuals from the elite group, create a new one by executing *crossover* on them and mutating the result. Two crossover operations are designed:

Crossover I: Let $\mathbf{p}_{BA(e1)}=[x_{BA(e1)}, y_{BA(e1)}, \theta_{BA(e1)}]^T$ and $\mathbf{p}_{BA(e2)}=[x_{BA(e2)}, y_{BA(e2)}, \theta_{BA(e2)}]^T$ denote the two elite individuals. Mix their position parts and orientation parts:

$$\begin{aligned}\mathbf{p}_{BA(\text{new})} &= \mathbf{crossover\_I}(\mathbf{p}_{BA(e1)}, \mathbf{p}_{BA(e2)}) \\ &= [x_{BA(e1)}, y_{BA(e1)}, \theta_{BA(e2)}]^T\end{aligned}$$

Crossover II: Make a random linear combination of the two individuals (λ is a randomly generated real value in [0, 1]):

$$\begin{aligned}\mathbf{p}_{BA(\text{new})} &= \mathbf{crossover\_II}(\mathbf{p}_{BA(e1)}, \mathbf{p}_{BA(e2)}) \\ &= \lambda \mathbf{p}_{BA(e1)} + (1-\lambda)\mathbf{p}_{BA(e2)}\end{aligned}$$

The *Crossover I* and *Crossover II* are specially designed crossover operations to directly handle the real-value vector form of $\mathbf{p}_{BA(k)}$. We can make an analogy between them and the traditional bit string based crossover operations by adopting the *Building Block Hypothesis* [39]. For $\mathbf{p}_{BA(k)}$, if we treat its position part and orientation part as basic building blocks, the traditional crossover operation turns to be *Crossover I*, see Fig.2-top. If we treat the proportion between its position part and orientation part as basic building block, the traditional crossover operation turns to be *Crossover II*, see Fig.2-bottom.

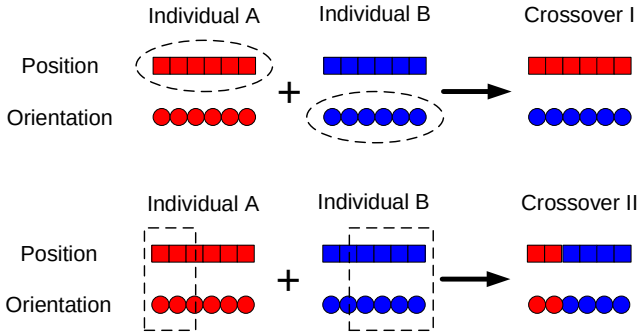

Fig.2. Crossover operation: (top) crossover I; (bottom) crossover II

It is worthy noting that the *Building Block Hypothesis* is a theory that heuristically explains the mechanism of GA. It can be rather treated as a guide for designing genetic operations.

(2-d-iii) *Re-initialization*: generate an individual as in *initialization*. This step is to keep the population diversity.

When two vehicles cooperate, the *initialization* is executed once and the *evolution* is iterated. A *dynamic* scheme of GA is used: the generation of $\mathbf{p}_{BA}$ from last period is propagated to the current period and evolved. We can assign only few times of evolution for a period (for example, once), which largely reduces the computational burden. As long as the vehicles are in cooperation, the *evolution* can be executed unceasingly, and the dynamic scheme of GA will finally find the optimal $\mathbf{p}_{BA}$. In our practice, the genetic evolution can usually find the optimal $\mathbf{p}_{BA}$ within only few periods (less than one second).

V. Indirect Vehicle-to-Vehicle Relative Pose Estimation

Given two vehicles A and B, the relationship between their local occupancy grid maps $\mathbf{M}_A$ and $\mathbf{M}_B$, i.e. $\mathbf{p}_{BA}$, can be established by the merging method previously introduced. The relationship between a vehicle and its local map, i.e. $\mathbf{S}_A$ or $\mathbf{S}_B$, is established by local SLAM. Thus, we can **indirectly** estimate the vehicle-to-vehicle relative pose $\mathbf{p}_{vBA}$ (that of B relative to A) via the chain relationship in (10):

$$\mathbf{p}_{vBA} = inv(\mathbf{S}_A) \oplus \mathbf{p}_{BA} \oplus \mathbf{S}_B \tag{10}$$

With this indirectly recovered $\mathbf{p}_{vBA}$, we can basically associate any specific perception (S.P.) of the two vehicles: given a point registered by vehicle B as $\mathbf{p}_{B(sp)}$, its counterpart in the view of vehicle A, i.e. $\mathbf{p}_{A(sp)}$, can be estimated via (11):

$$\mathbf{p}_{A(sp)} = \mathbf{p}_{vBA} \oplus \mathbf{p}_{B(sp)} = inv(\mathbf{S}_A) \oplus \mathbf{p}_{BA} \oplus \mathbf{S}_B \oplus \mathbf{p}_{B(sp)} \tag{11}$$

Therefore, this strategy of indirect vehicle-to-vehicle relative pose estimation can serve as a general solution for multi-vehicle perception association, as illustrated in Fig.3.

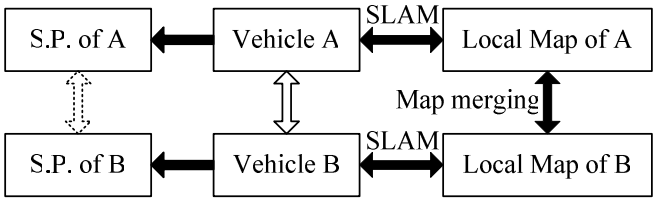

Fig.3. Indirect vehicle-to-vehicle relative pose estimation.

In other words, even for applications where local occupancy grid maps are not directly needed, we can still let the vehicles to carry out the method presented in section III, section IV, and this section, in order that the vehicles can effectively estimate their relative pose for perception association purpose.

VI. Experiment

A. Experimental Conditions

Experiments were carried out in INRIA campus, based on two CyCab vehicles developed by the Imara lab. A CyCab is equipped with a RTK-GPS, an IBEO 2-D laser scanner, odometers (including a steering encoder), and a wireless router (Laguna box from the Commsignia company), see Fig.4.

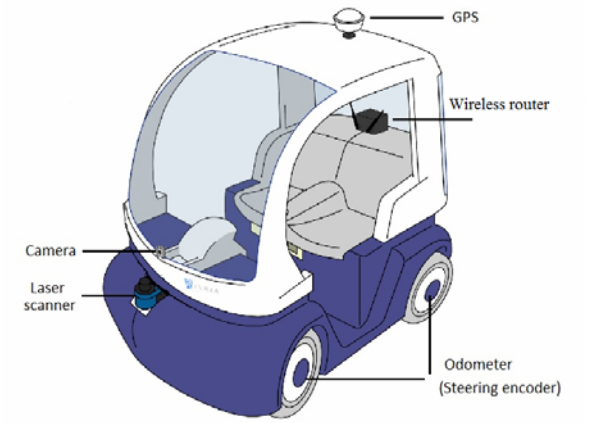

Fig.4. Intelligent vehicle system configurations

The vehicles were synchronized with the GPS universal time—even a normal GPS provides accurate universal time. For a vehicle, the RTK-GPS outputs were used to maintain an accurate estimate of the vehicle global state, denoted as $\mathbf{X}_{(RTK)}$, which was only treated as the ground-truth. For experiments, we deliberately used degraded GPS outputs, as if the vehicle was equipped with a normal GPS.

B. Occupancy Grid Map Merging: Ground-Truth

For a pair of local occupancy grid maps ($\mathbf{M}_A$ and $\mathbf{M}_B$) built by the two CyCab vehicles (A and B), we can estimate their alignment via (12):

$$\mathbf{p}_{BA(RTK)} = \mathbf{S}_A \oplus inv(\mathbf{X}_{A(RTK)}) \oplus \mathbf{X}_{B(RTK)} \oplus inv(\mathbf{S}_B) \qquad (12)$$

$\mathbf{X}_{A(RTK)}$, $\mathbf{X}_{B(RTK)}$, $\mathbf{S}_A$, and $\mathbf{S}_B$ are rather accurate, yet there still exists certain error in each of them. Thus, there also exists certain error in $\mathbf{p}_{BA(RTK)}$, which causes slight inconsistency between the two maps. To refine $\mathbf{p}_{BA(RTK)}$, we perform a dense searching in a small range around $\mathbf{p}_{BA(RTK)}$ and choose the one with highest fitness value as the ground-truth. Given a pair of local maps shown in Fig.5-left as an example, their merging effect based on $\mathbf{p}_{BA(RTK)}$ and refined $\mathbf{p}_{BA(RTK)}$ (ground-truth) are shown respectively in Fig.5-bottom-right and Fig.5-top-right: slight inconsistency between the two maps exists in Fig.5-bottom-right, whereas it is corrected in Fig.5-top-right.

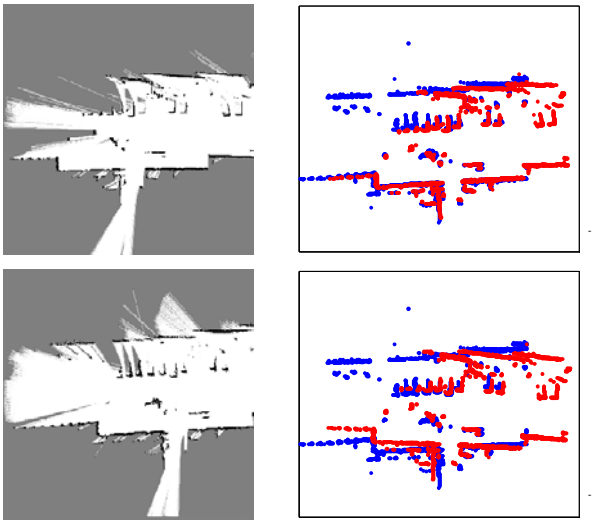
Fig.5. Ground-truth of the map alignment: (left) two occupancy grid maps; (top-right) the ground-truth; (bottom-right) slight inconsistency

C. Occupancy Grid Map Merging: Experiment I

The introduced occupancy grid map merging method was tested on totally 1155 pairs of local occupancy grid maps built by the two CyCab vehicles. In a real application, the GA can be realized in a dynamic way as described in section IV. In this experiment, however, we tested each pair independently without using the dynamic scheme, in order to examine the performance of the GA based optimization. For each pair, a new *initialization* (the population size is 1000) was executed; *evolution* was iterated until certain criteria were satisfied, i.e. the evolution result is within 20 cm (grid cell size) in position and 0.5° in orientation around the ground-truth.

Some map merging examples are demonstrated in Fig.6 to Fig.8. In each figure, the left two sub-figures display a pair of local occupancy grid maps (of size 80×80 m^2) built by the two CyCab vehicles. The bottom-right one displays the erroneous initial map alignment—to verify the method robustness, we deliberately exaggerated the initial map alignment error—The top-right one displays the correct merging.

The proposed method successfully merged all the pairs. For each pair, we compute the *convergence evolution number* (CEN), i.e. the number of genetic evolution needed for the optimization to converge. The CEN histogram for all the pairs is shown in Fig.9; the optimization normally converges within ten times of genetic evolution (less than one second).

The average CEN is 5.46. We can make a rough analysis for the computational complexity of the GA based optimization and that of an exhaustive searching (ES) based optimization (i.e. traversing all the possibilities for certain orientation and position resolution). We treat the fitness computation of an individual as one operation. A genetic evolution round takes

averagely about 1500 operations; then the average operation number for optimization convergence is about 5.46×1500≈8K. To guarantee the searching accuracy of 20 cm in position and 0.5° in orientation (the same as for the GA) within the initial error range of ±30 m in position and ±30° in orientation, the ES needs (60×60×60)/(0.2×0.2×0.5)=10800K operations. Thus, the GA based optimization is more efficient than the ES based optimization by a factor of 10800K/8K=1350.

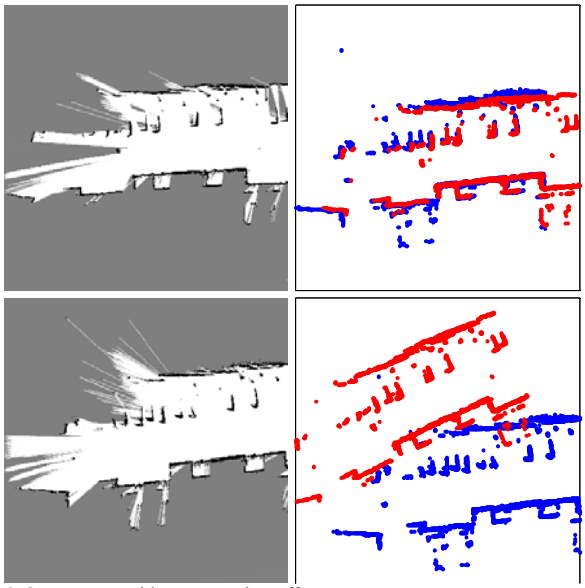

Fig.6. Occupancy grid map merging effect

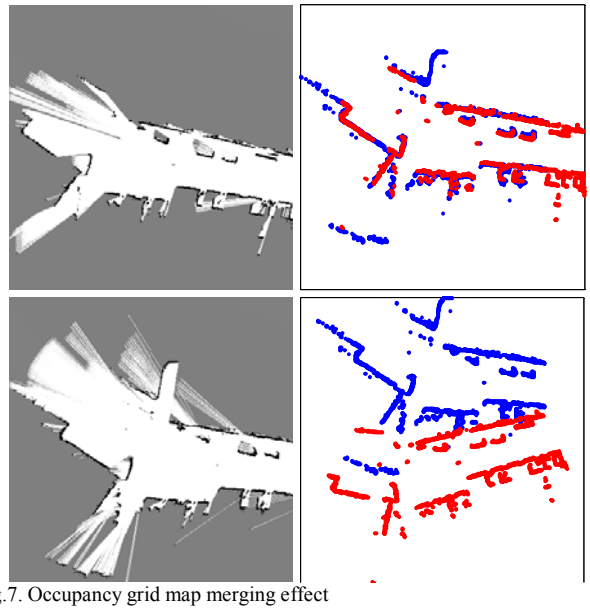

Fig.7. Occupancy grid map merging effect

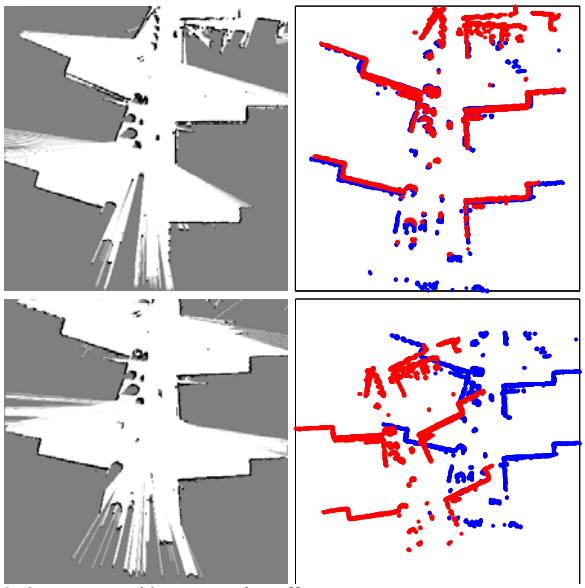

Fig.8. Occupancy grid map merging effect

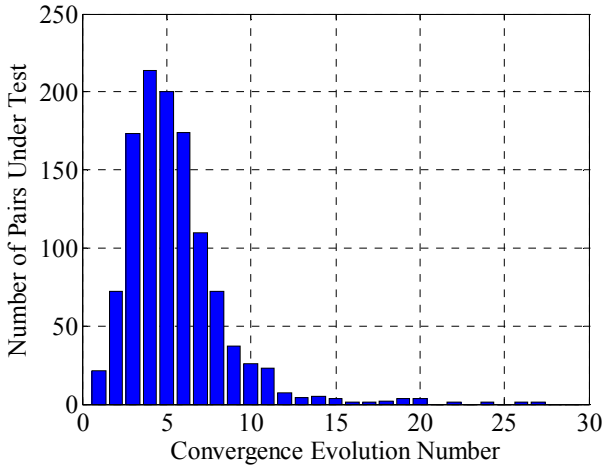

Fig.9. Histogram of the convergence evolution number

D. Occupancy Grid Map Merging: Experiment II

In this experiment, we demonstrate how the proposed occupancy grid map merging method can naturally recover the merging result from a *kidnapping* situation. Given a pair of local occupancy grid maps, as shown in Fig.10; first, they were merged using the proposed method. The process of genetic evolution is illustrated in Fig.11; in each sub-figure, the black points display the position part of the genetic individuals; the maps were aligned according to the individual with temporarily highest fitness value.

At the beginning, as shown in Fig.11-top-left, a population of individuals was initialized randomly in the error range—the initial map alignment can be rather erroneous. The Fig.11-top-right and Fig.11-bottom-left display the evolution results before convergence. Finally, after seven evolution rounds, the correct alignment (optimal individual) was found and the evolution result is shown in Fig.11-bottom-right; the population is more concentrated around the optimal one, whereas the diversity of the population is still maintained.

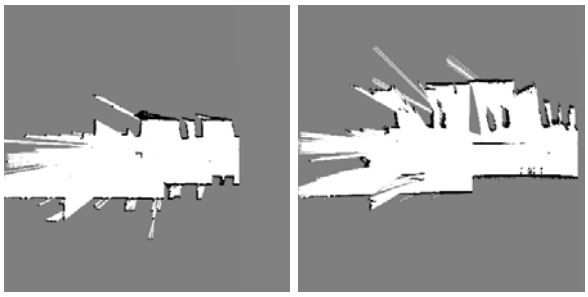

Fig.10. A pair of occupancy grid maps

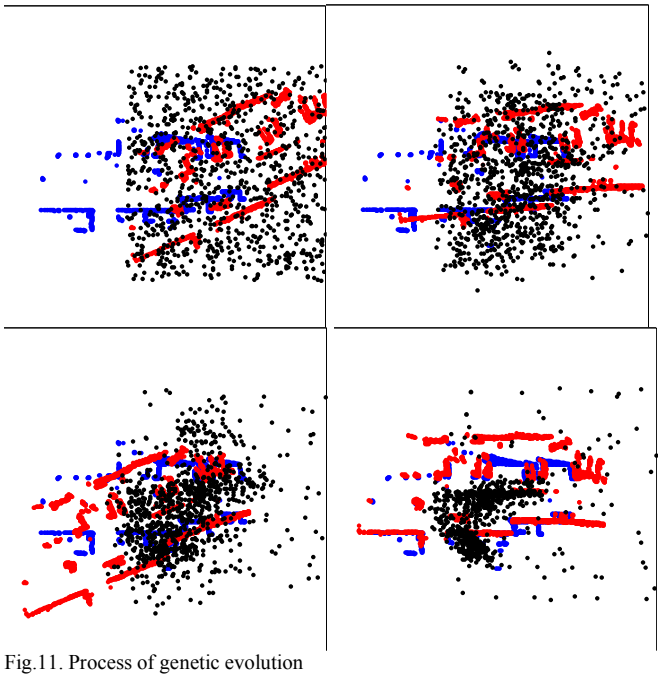

Fig.11. Process of genetic evolution

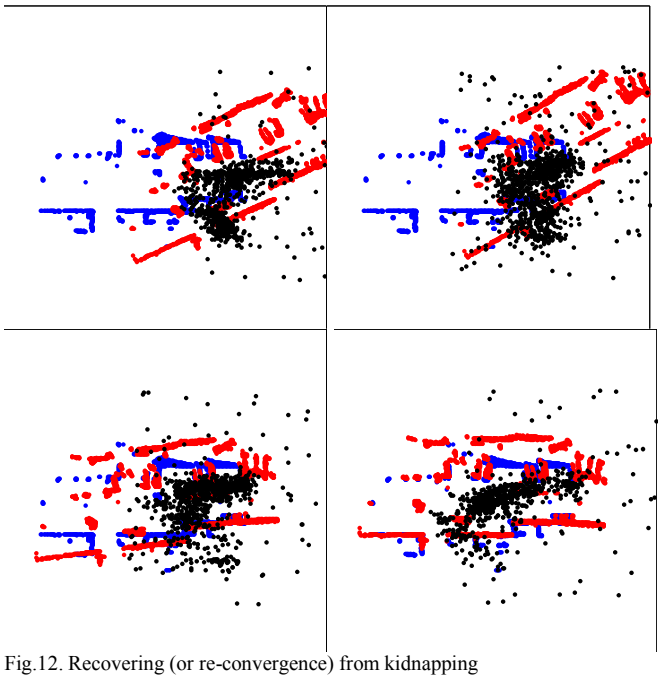

Fig.12. Recovering (or re-convergence) from kidnapping

Then we displaced the optimal individual back to the initial erroneous point in the solution space and displaced all other individuals accordingly. We did not provide any clue of this change. This is like that the entire population was kidnapped to a wrong place, without being informed how they had been displaced or even whether they had been displaced; see the change from Fig.11-bottom-right to Fig.12-top-left.

For this population kidnapped, the genetic evolution continued in the same way as if no kidnapping had happened, as illustrated in Fig.12. After nine evolution rounds, the population recovered from the kidnapping, and the correct solution was found again, as shown in Fig.12-bottom-right.

A kidnapping hardly happens in reality, yet this experiment demonstrates the potential of the proposed method to recover the merging result from unexpected misleading factors.

E. Occupancy Grid Map Merging: Experiment III

An example is illustrated in Fig.13; the top two sub-figures display a pair of local occupancy grid maps built by the two CyCab vehicles. There is considerable inherent inconsistency between the two maps. To merge these two maps, we used the objective function in (9) (set α to 0.01) and optimized it also with the GA. After some evolution rounds, an optimal solution was found, as shown in Fig.13-bottom-right, which is wrong. We computed the fitness value of the ground-truth via (9). In fact, based on (9), the fitness value of the ground-truth is indeed smaller than that of the wrong optimal solution. This shows that the wrong optimal solution was not caused by mal-functioning of the GA, but was caused by use of (9). In contrast, the proposed method using (8) succeeded in finding the correct solution, as shown in Fig.13-bottom-left.

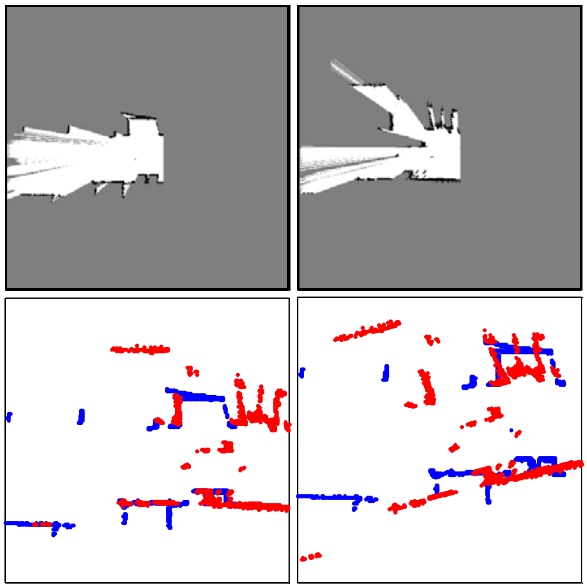

Fig.13. Map inherent inconsistency: (top) two local maps; (bottom-left) correct map merging; (bottom-right) wrong map merging

As demonstrated by this example, use of (9), which requires tuning the heuristic parameter α to balance the influence of the distance term, can cause false merging if α is not correctly tuned. In contrast, use of (8), which totally neglects the distance term and is exempt from tuning α, can always

succeed in the merging task (as also demonstrated in previous sub-sections). For the spirit of *Occam's razor*, we naturally prefer (8) instead of (9).

F. Applications of the Indirect Vehicle-to-Vehicle Relative Pose Estimation Strategy

The effectiveness of the proposed occupancy grid map merging method contributes to the effectiveness of the indirect vehicle-to-vehicle relative pose estimation strategy given in section V. We demonstrate several application examples of this strategy briefly—the application details are omitted here, as they are out of the focus of this paper.

For example, this strategy is useful for multi-vehicle cooperative localization [40]. This strategy can also be used to associate the detected moving objects from two vehicles, as shown in Fig.14. Each of the left two sub-figures shows the local occupancy grid map and the detected moving objects (marked by blue boxes) of a vehicle. The merged map and moving objects are shown in Fig.14-bottom-right.

This strategy can also be used to relate the visual data (if existing) of two vehicles, forming an effect coined as *cooperative augmented reality* (CAR), as shown in Fig.15 (also see Fig.1). Fig.15 shows an augmented view of the second vehicle; we can notice the 3D vividness of the augmented effect, as if we can directly see through the occlusion caused by the first vehicle. These effects shown in Fig.14 and Fig.15 will be valuable for enhancing the driving safety and efficiency of the second vehicle in a challenging scenario as shown in Fig.1.

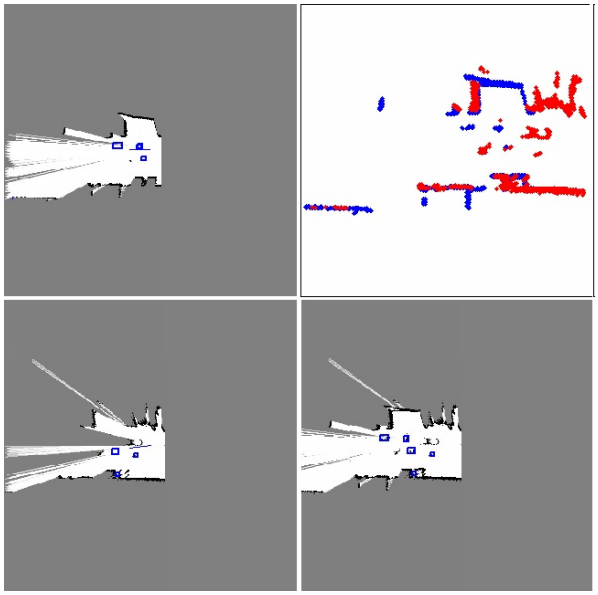

Fig.14. Cooperative moving object detection: (left) local maps and moving object detection; (top-right) local map merging; (bottom-right) merged moving objects

Fig.15. Cooperative augmented reality (CAR): 'see' through the occlusion

VII. CONCLUSION

In this paper, we have introduced a method of occupancy grid map merging, dedicated to multi-vehicle cooperative local mapping purpose in outdoor environment. In a general map merging framework, we have proposed an objective function based on occupancy likelihood, and provided some concrete procedures designed in the spirit of genetic algorithm to optimize the defined objective function. Based on this method, we have further described a strategy of indirect vehicle-to-vehicle relative pose estimation, which can serve as a general solution for multi-vehicle perception association.

We have presented a variety of experiments on the proposed occupancy grid map merging method. Experiments have shown that the proposed method can effectively fulfill the merging task, even facing an exaggerated initial map alignment error and considerable map inherent inconsistency. The proposed method can naturally recover the merging result from a *kidnapping* situation, which demonstrates its potential to recover the merging result from unexpected misleading factors. The effectiveness of the proposed occupancy grid map merging method contributes to the effectiveness of the indirect vehicle-to-vehicle relative pose estimation strategy; we have briefly demonstrated several useful application examples of this strategy, such as cooperative moving object detection, and cooperative augmented reality.

The presented practice of cooperative local mapping will be valuable for enhancing the driving safety and efficiency of vehicles operating in some challenging scenarios. Meanwhile, the presented works still allow for further improvements. For example, although we have presented a comparative study that explains why we did not use a distance term as in literature, yet it is a pity that the distance information is totally neglected in our proposed objective function. How to reasonably take the distance information into account in the objective function is a point for future research. For another example: arbitrarily given a map merging method, how to design a generally effective way to automatically judge the map merging correctness is still a difficult open question—existing works in literature always rely on certain *ad hoc* empirical rules that are not generally applicable—For the presented works, this

question is also a point that deserves further research.

APPENDIX: COORDINATE TRANSFORM

We follow the compounding notation in [10]:

$$\begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} \oplus \begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_2\cos\theta_1 - y_2\sin\theta_1 + x_1 \\ x_2\sin\theta_1 + y_2\cos\theta_1 + y_1 \\ \theta_2 + \theta_1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} \oplus \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_2\cos\theta_1 - y_2\sin\theta_1 + x_1 \\ x_2\sin\theta_1 + y_2\cos\theta_1 + y_1 \end{bmatrix}$$

$$inv\left(\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}\right) = \begin{bmatrix} -x\cos\theta - y\sin\theta \\ x\sin\theta - y\cos\theta \\ -\theta \end{bmatrix}$$

Given an occupancy grid map $\mathbf{M}=\{(\mathbf{c}_1,o_1), (\mathbf{c}_2,o_2), \ldots, (\mathbf{c}_n,o_n)\}$, where $\mathbf{c}_i$ and o_i respectively denote the position and the occupancy state of the i-th cell; given $\mathbf{p}=[x,y,\theta]^T$, define:

$$\mathbf{p} \oplus \mathbf{M} = \{(\mathbf{p} \oplus \mathbf{c}_1, o_1), (\mathbf{p} \oplus \mathbf{c}_2, o_2), \ldots, (\mathbf{p} \oplus \mathbf{c}_n, o_n)\}$$

REFERENCES

[1] I. Skog, P. Handel, "In-car positioning and navigation technologies-a survey", *IEEE Trans on Intelligent Transportation Systems*, vol.10, no.1, pp.4-21, 2009

[2] J. Laneurit, R. Chapuis, F. Chausse, "Accurate vehicle positioning on a numerical map", *Int Journal of Control, Automation, & Systems*, vol.3, no.1, pp.15-31, 2005

[3] M. Enzweiler, D.M. Gavrila, "Monocular pedestrian detection: survey and experiments", *IEEE Trans on Pattern Analysis & Machine Learning*, vol.31, no.12, pp.2179-2195, 2009

[4] M. Bertozzi, A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection", *IEEE Trans on Image Processing*, vol.7, no.1, pp.62-81, 1998

[5] D.A. Pomerleau, "ALVINN: An autonomous land vehicle in a neural network", *Tech Report CMU-CS-89-107*, Carnegie Mellon Univ., 1989

[6] R. Rajamani, "Vehicle dynamics and control", *New York: Springer-Verlag*, 2005

[7] H. Durrant-Whyte, T. Bailey, "Simultaneous localization and mapping: part I", *IEEE Robotics & Automation Magazine*, vol.13, no.2, pp.99-110, 2006

[8] T. Bailey, H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II", *IEEE Robotics & Automation Magazine*, vol.13, no.3, pp.108-117, 2006

[9] G. Grisetti, R. Kummerle, C. Stachniss, W. Burgard, "A tutorial on graph-based SLAM", *IEEE Intelligent Transportation Systems Magazine*, vol.2, no.4, pp.31-43, 2010

[10] C.C. Wang, "Simultaneous Localization, Mapping and Moving Object Tracking", *PhD Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh PA*, 2004

[11] T.D. Vu, "Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects", *PhD Thesis, Institut National Polytechnique de Grenoble*, 2009

[12] ISO 21217:2010 Intelligent transport systems-Communications access for land mobiles (CALM)-Architecture, April 2010

[13] Intelligent Transport Systems (ITS): Communications Architecture, September 2010. ETSI EN 302 665 V1.1.1 (2010-09)

[14] J.S. Gutmann, C. Schlegel, "AMOS: Comparison of scan matching approaches for self-localization in indoor environments", *Proc 1st Euromicro Workshop on Advanced Mobile Robot*, 1996, pp.61-67

[15] F. Lu, E. Milios, "Globally consistent range scan alignment for environment mapping", *Autonomous Robots*, vol.4, pp.333-349, 1997

[16] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem", *Proc AAAI National Conf on Artificial Intelligence*, 2002, pp.593-598

[17] J. Guivant, E. Nebot, H. Durrant-Whyte, "Simultaneous localization and map building using natural features in outdoor environments", *IAS-6 Intelligent Autonomous Systems*, 2000, pp.581-586

[18] I. Cox, "Blanche – An experiment in guidance and navigation of an autonomous robot vehicle", *IEEE Trans on Robotics & Automation*, vol.7, no.2, pp.193-204, 1991

[19] E. Royer, "Cartographie 3D et localisation par vision monoculaire pour la navigation autonome d'un robot mobile", *PhD Thesis, Université Blaise Pascal-Clermont II*, 2006

[20] A. Gil, O. Reinoso, M. Ballesta, M. Julia, "Multi-robot visual SLAM using a Rao-Blackwellized particle filter", *Robotics & Autonomous Systems* 58, pp.68-80, 2010

[21] A. Elfes, "Occupancy grids: a probabilistic framework for robot perception and navigation", *PhD Thesis, Carnegie Mellon University*, 1989

[22] R. Madhavan, K. Fregene, L.E. Parker, "Distributed cooperative outdoor multirobot localization and mapping", *Autonomous Robots*, vol.17, no.1, pp.23-39, 2004

[23] A. Howard, L.E. Parker, G.S. Sukhatme, "The SDR experience: Experiments with a large-scale heterogeneous mobile robot team", *Experimental Robotics IX, STAR 21, Springer-Verlag Berlin Heidelberg*, 2006, pp.121-130

[24] E.D. Nerurkar, S.I. Roumeliotis, A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization", *IEEE Int Conf on Robotics & Automation*, 2009, pp.1402-1409

[25] L. Carlone, M.K. Ng, J. Du, B. Bona, M. Indri, "Simultaneous localization and mapping using rao-blackwellized particle filters in multi robot systems", *Journal of Intelligent & Robotic Systems*, vol.63, no.2, pp.283-307, 2011

[26] D. Fox, W. Burgard, H. Kruppa, S. Thrun, "A probabilistic approach to collaborative multi-robot localization", Autonomous Robots, 8(3), 2000, pp.325-344

[27] A. Howard, M.J. Mataric, G.S. Sukhatme, "Putting the 'I' in 'team': an ego-centric approach to cooperative localization", *IEEE Int Conf on Robotics & Automation*, vol.1, 2003, pp.868-874

[28] F. Lu, E. Milios, "Robot pose estimation in unknown environments by matching 2D range scans", *Journal of Intelligent & Robotic Systems*, vol.18, pp.249-275, 1997

[29] A. Birk, S. Carpin, "Merging occupancy grids from multiple robots", *Proc IEEE*, vol.94, no.7, pp.1384-1397, 2006

[30] A. Grossmann, R. Poli, "Robust mobile robot localization from sparse and noisy proximetry readings using Hough transform and probability grids", *Robotics & Autonomous Systems*, vol.37, pp.1-18, 2001

[31] G. Dedeoglu, G. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots", *Int Symp on Distributed Autonomous Robotics Systems*, 2000, pp.251-260

[32] S. Topal, I. Erkmen, A.M. Erkmen, "A novel map merging methodology for multi-robot systems", *Proc World Congress on Engineering & Computer Science*, 2010, pp.383-387

[33] S. Saeedi, L. Paull, M. Trentini, H. Li, "Multiple Robot Simultaneous Localization and Mapping", *IEEE/RSJ Int Conf on Intelligent Robots & Systems*, 2011, pp.853-858

[34] P.J. Besl, N.D. McKay, "A method for registration of 3-D shapes", *IEEE Trans on Pattern Analysis & Machine Intelligence*, vol.14, no.2, pp.239-256, 1992

[35] Y. Chen, G.G. Medioni, "Object modeling by registration of multiple range images", *Image & Vision Computing*, vol.10, no.3, pp.145-155, 1992

[36] S. Rusinkiewicz, M. Levoy, "Efficient variants of the ICP algorithm", *3rd Int Conf on 3D Digital Imaging & Modeling*, 2001, pp.145-152

[37] J. Minguez, L. Montesano, F. Lamiraux, "Metric-based Iterative Closest Point scan matching for sensor displacement estimation", *IEEE Trans on Robotics*, vol.22, no.5, pp.1047-1054, 2006

[38] P. Biber, W. Strasser, "The normal distributions transform: A new approach to laser scan matching", *IEEE/RSJ Int Conf on Intelligent Robots & Systems*, 2003, pp.2743-2748

[39] K.F. Man, K.S. Tang, S. Kwong, "Genetic algorithms", *New York: Springer-Verlag*, 1999

[40] H. Li, F. Nashashibi, "Multi-vehicle cooperative localization using indirect vehicle-to-vehicle relative pose estimation", *IEEE Int Conf on Vehicular Electronics & Safety*, 2012, pp.267-272