

# Merging Maps via Hough Transform

Stefano Carpin

**Abstract**—We present a recently developed algorithm for merging multiple occupancy grid maps computed by multiple robots independently exploring a shared indoor environment. The algorithm exploits the well known Hough transform in a novel way in order to produce a set of ranked roto-translations aimed to overlap the partial maps provided as input. In this paper, after having briefly summarized such method, we investigate the impact on the performance of different variations of the Hough transform. In particular, we are interested in determining the repercussions in terms of accuracy and computational time when only a subset of points is used to compute the transformation. Results are analyzed while merging maps produced by two robots exploring an indoor environment, and also using public available data sets. It turns out that the proposed method is robust and positively influenced by the use of more refined approaches to compute the Hough transform.

## I. INTRODUCTION

Many reasons stimulate research in multi-robots systems. Among them we find the possibility to build systems that are more robust, efficient, cost-effective, and usable than single robots [1]. These benefits come however at their own cost, as the development of multi-robot systems offers challenges and problems not found when implementing solutions based on a single robot. One of the many difficulties concerns sensor fusion. Multi-robot systems are distributed systems collecting information distributed both spatially and in time, and a successful coordination between robots often relies on correctly fusing together these different data sources.

In this paper we address one instance of this problem, namely the integration of planar occupancy grid maps built by robots exploring different parts of the same environment. When multiple robots are used to explore an unknown area two different approaches can be envisioned. A first alternative consists in treating the various robots as the components of a unique system collecting information at different locations and performing data fusion instantaneously. Two major drawbacks affect this method. First the overall dimensionality of the space increases, and secondly it is necessary to assume robots can constantly communicate with each other. As an alternative, one can assume each robot operates individually, and information is exchanged and integrated only during occasional meetings between robots.

This later approach is embraced in this paper, and we further develop a former line of research we developed in the past, as detailed in section II. In particular, we recently developed a new method [2] that combines multiple maps by exploiting the *Hough spectrum* concept, i.e. a cross product of the well known Hough transform. The method is novel in

many aspects, and presents various advantages with respect to formerly developed algorithms:

- 1) it produces a set of ranked hypothesis rather than a single one, thus enabling to deal with ambiguous scenarios without committing too early to a single assumption when more data are needed to disambiguate the situation;
- 2) the algorithm is fast and capable of merging maps in far less time than our formerly developed iterative approaches;
- 3) differently from other solutions proposed in the past, the algorithm never requires that robots determine their mutual positions in the partial maps they build and merge.

Since our goal is to perform map fusion frequently, it is extremely important to obtain reliable results in short time, and with this respect a vanilla Hough transform implementation leaves room to plenty of improvements. In this manuscript we experimentally evaluate some algorithmic improvements to Hough transform computation and we verify their impact on the technique we propose. In order to set the right framework for our contribution, it is necessary to outline that since we use the Hough transform to detect straight lines, we will assume the maps to be merged present many linear features. This is compatible with the hypothesis that robots are exploring indoor environments.

The paper is organized as follow. After having detailed about related work in section II, our novel method and possible improvements are presented in section III. Experimental results are detailed in section IV and conclusions are drawn in section V.

## II. RELATED WORK

Autonomous mapping is one of the topics attracting the greatest amount of interest from robotics researchers, and the amount of related literature is huge. We refer the reader to the recent book by Thrun et al. [3] for a comprehensive survey on the topic. Research in map merging is instead much more limited, and this will be the thread of this section. Among the few papers in the area we find our previous contributions [4][5], where map merging was attacked as an optimization problem. An iterative search in the space of possible rigid transformations was performed with the goal of maximizing a target overlapping function. The iterative nature of this approach makes it unsuitable for online systems, where maps have to be frequently merged during the mission, rather than after the mission. Howard et al. studied the online problem in [6], and developed an algorithm where robots exchange and merge maps when they meet. Their system implies that

robots localize each other in their respective maps before merging. If reciprocal localization requires visual line of sight, this method is inapplicable in situations where robots exchange information without seeing each other. A similar technique was developed by Fox and colleagues in [7]. Contrarily to the previous contribution, robots do not meet randomly during the mission, but their meetings are planned in order to increase the overall quality of the map being produced. In this approach robots also mutually localize each other in their maps, but this is done by exchanging sensor data. Similar techniques, i.e. systems where mutual localization is needed, were also proposed in [8][9][10]. A radically different algorithm was presented by Huang and Beevers [11]. Their idea differs in that it assumes that maps are represented as topological maps rather than by occupancy grid maps. Starting from this representation, they look for the common subgraphs between the maps being merged in order to identify possible merging hypothesis. The main problem of this interesting approach is in the inherent complexity of the problem of detecting common subgraphs.

Techniques similar to map merging have been developed in machine vision and are known as image registration, alignment or stitching. The reader is referred to [12] for a recent survey on the topic.

### III. MERGING MAPS VIA HOUGH TRANSFORM

In the following a grid map  $M$  is represented by a matrix with  $r$  rows and  $c$  columns. Each cell  $M(i, j)$  may contain three values, indicating whether the cell is *traversable*, *occupied*, or if its status is *unknown*. SLAM algorithms commonly produce beliefs for these three values for each grid cell, so our model is compatible with the state of the art. Given two maps,  $M_1$  and  $M_2$ , the goal of map merging is to find a rigid transformation  $T$  overlapping them. The transformation  $T$  is the combination of a rotation  $\psi$ , followed by a translation along the  $x$  and  $y$  axis of magnitude  $\Delta x$  and  $\Delta y$ , respectively. To make explicit the dependence of the transformation  $T$  on these three parameters, we will write  $T(\Delta x, \Delta y, \psi)$ . Moreover, we write  $TM$  for the map obtained from map  $M$  after applying transformation  $T$ <sup>1</sup>. From now onwards, whenever we consider two maps being merged, we will tacitly assume they have the same number of rows and columns. This assumption simplifies the algorithmic description and can be easily accommodated during the implementation stage by properly padding the maps if needed. Given two maps, multiple transformations overlapping them can be determined, therefore a metric to decide which one is better is needed. In order to rank different transformations we use the *acceptance index* we formerly introduced [4][5]. The acceptance index considers the *agreement* and *disagreement* between two maps. The former is the number of cells in  $M_1$  and  $M_2$  that are both free or both occupied, while the latter is the number of cells such that  $M_1$  is free

and  $M_2$  is occupied or vice-versa. Indicating the former as *agr* and the latter as *dis*, the acceptance index is defined as follows:

$$\omega(M_1, M_2) = \begin{cases} 0 & \text{if } agr = 0 \\ \frac{agr(M_1, M_2)}{agr(M_1, M_2) + dis(M_1, M_2)} & \text{if } agr \neq 0 \end{cases} \quad (1)$$

The reader should note that  $\omega(M, M) = 1$ , and that the definition does not consider cell elements in  $M_1$  or  $M_2$  whose state is unknown. Given two possible transformations  $T_1$  and  $T_2$ , we prefer  $T_1$  if  $\omega(M_1, T_1 M_2) > \omega(M_1, T_2 M_2)$ . One instance of the map merging problem therefore consists of two maps,  $M_1$  and  $M_2$ , and the goal is to determine a transformation  $T$  that maximizes  $\omega(M_1, TM_2)$ . The map merging problem can therefore be seen as an optimization problem where the transformation achieving the best acceptance index is sought. In the method we propose in this paper the optimal transformation is searched in two steps, i.e. first the rotation  $\psi$  is determined, and then the displacements  $\Delta x$  and  $\Delta y$  are computed.

#### A. The Hough spectrum and its applications of map merging

Censi et al. [13] introduced the concept of Hough spectrum for the problem of scan matching. Their idea is the building block for the step estimating the rotation  $\psi$ . Hough transform is a well known method to detect lines in binary images [14]. The method exploits the fact that lines can be represented through the polar representation. This means that a line in the  $x$ - $y$  plane can be described as the locus of points satisfying the relationship

$$\rho = x \cos \theta + y \sin \theta \quad (2)$$

where  $\rho$  is the distance of the line from the origin, and  $\theta$  is the angle between the  $x$  axis and the normal from the line to the origin. For implementation efficiency, line detection is performed using the Discretized Hough transform (DHT). In DHT a subset of the  $\rho$ - $\theta$  plane is divided into a grid of accumulators with  $n_\theta$  columns and  $n_\rho$  rows. All accumulators are initially set to 0. If the binary image consists of  $m$  points, the sinusoid defined by equation 2 is computed for each of the  $m$  points, and the accumulators intersected by the associated curve are incremented. After this step, maxima are found where multiple sinusoids intersect. Each maximum is associated with a specific  $\rho, \theta$  couple, thus identifying a line in the image. DHT can be applied to detect lines in an occupancy grid map  $M$  by converting it into a binary image. All occupied cells are set to black, while traversable and unknown cells are set to white. Once the Hough transform has been computed, the Hough spectrum is defined as the vector with  $n_\theta$  elements obtained by adding columnwise the squared values of the Hough transform. To be more specific, if the result of DHT is stored in a matrix  $\mathcal{H}$ , with  $n_\theta$  columns and  $n_\rho$  rows, then its Hough spectrum is defined by the following relationship:

$$\mathcal{HS}(k) = \sum_{i=1}^{n_\rho} \mathcal{H}(i, k)^2 \quad 1 \leq k \leq n_\theta$$

<sup>1</sup>This notation has a convenient matrix interpretation that is here omitted for lack of space. While such interpretation is rather straightforward, the reader is referred to our cited former papers for more details concerning this aspect.

Informally speaking, the Hough spectrum provides indications about directions more common among the lines in the image. Cross correlation between two Hough spectra reveals how one of the two signals should be translated in order to maximize the overlap between the two. Since Hough spectra carry information about orientations, the translation identified by the cross correlation indicates the rotation needed to align the two maps. Figure 1 shows two Hough spectra and their cross correlation. It is possible to observe that Hough spectra, as well cross correlation, exhibit a 180 degrees periodicity. Therefore cross correlation between Hough spectra has to be performed using a circular approach, with the final extreme of the spectrum *wrapping back* towards the beginning. Multiple local maxima can be observed in the third panel of

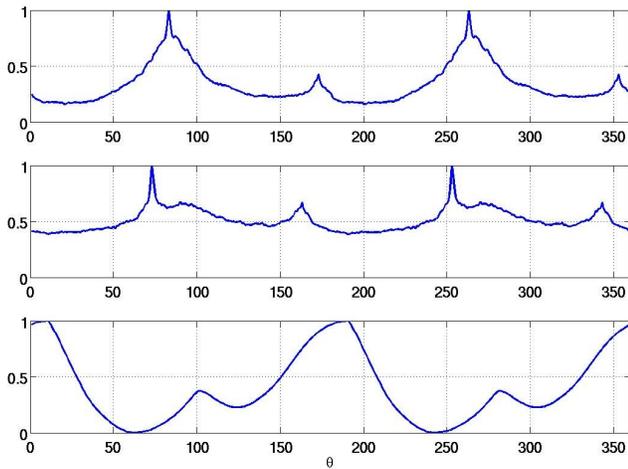


Fig. 1. The top two panels show two Hough spectra computed from two maps, while the bottom shows their cross correlation. The 180 degrees periodicity can be appreciated in these Hough spectra examples.

figure 1. Each of them corresponds to a potential rotation  $\psi_i$  aligning the two maps. The algorithm we are describing can be track each of them.

### B. Displacement computation

Let  $\psi_i$  be one local maxima extracted from the cross correlation between the Hough spectra of  $M_1$  and  $M_2$ , and let  $M'_2$  be the map obtained from  $M_2$  after rotation  $\psi_i$ , i.e.  $M'_2 = T(0, 0, \psi_i)M_2$ . Translations  $\Delta_x^i$  and  $\Delta_y^i$  associated with  $\psi_i$  can be determined by means of a bidimensional correlation between  $M_1$  and  $M'_2$ . This approach is however computationally expensive and can be avoided exploiting correlations once again. The *X-spectrum* and *Y-spectrum* of a binary image  $M$  obtained from an occupancy grid map are two vectors with  $c$  and  $r$  components respectively defined as follows:

$$\mathcal{S}_x^M(j) = \sum_{i=1}^r M(i, j) \quad 1 \leq j \leq c \quad (3)$$

$$\mathcal{S}_y^M(i) = \sum_{j=0}^c M(i, j) \quad 1 \leq i \leq r \quad (4)$$

These vectors are the *projections* along the  $x$  and  $y$  axis of the two images. Given the  $X$  and  $Y$ -spectra of two aligned images,  $\Delta_x$  and  $\Delta_y$  can be also easily computed through cross correlation. This step is illustrated in figure 2, where an occupancy grid map is displayed together with its associated  $X$ -spectrum. It is possible to observe four distinctive peaks in the spectrum, two on the left and two on the right. The first two are associated with the walls surrounding the wide hall on the left of the map, while the other two are generated by the vertical corridor situated on the right of the map. Additional peaks are found in correspondence with other features in the map. This *signature* eases the determination of the needed translations by mean of cross correlation between spectra. A similar technique was used in [15].

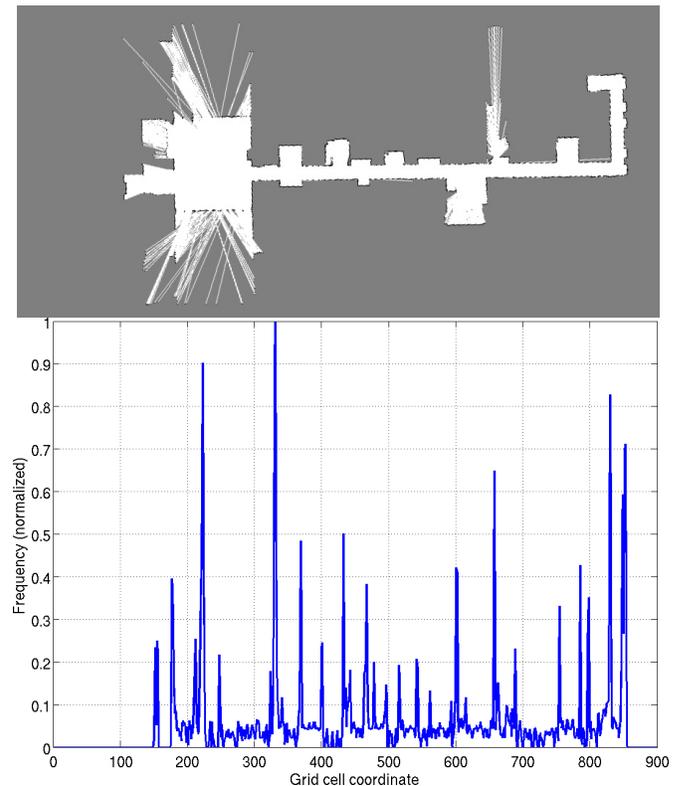


Fig. 2. The top panel shows one map built during the experiments described in section IV. The bottom panel shows the  $X$ -spectrum of the map. As described in the paper, the image was preliminarily aligned with the axis in order to fully exploit the  $X$ -spectrum. Spectrum values were normalized in the range 0-1.

The approach may be brittle if the maps do not provide distinctive projections along the  $x$  and  $y$  axis. However, since the focus of this research is about merging maps of building interiors, we can exploit the fact that walls in buildings are not randomly arranged, but they are most often aligned with two orthogonal directions. The potential brittleness of the approach based on the  $X$  and  $Y$  spectrum can therefore be easily overcome by aligning one of the two maps with the axis before starting the overall merging algorithm. This step is straightforward to achieve because the required rotation can be revealed by its Hough spectrum. Therefore from now

on we assume that one of the two maps has been preliminary aligned with the axis.

### C. Algorithmic sketch

The idea illustrated in the former two subsections can be easily turned into the algorithm illustrated in 1. The algorithm processes two occupancy grid maps,  $M_1$  and  $M_2$  and produces a set of transformation hypothesis. It is assumed that the number of hypothesis  $n$  is one of the parameters accepted by the algorithm.

---

#### Algorithm 1 Details

---

- 1: **ComputeHypothesis**( $M_1, M_2, n$ )
  - 2:  $\mathcal{HS}_1 \leftarrow \text{HoughSpectrum}(M_1)$
  - 3:  $\mathcal{HS}_2 \leftarrow \text{HoughSpectrum}(M_2)$
  - 4:  $\mathcal{C} \leftarrow \text{CrossCorrelation}(\mathcal{HS}_1, \mathcal{HS}_2)$
  - 5:  $\psi_1 \dots \psi_n \leftarrow \text{LocalMaxima}(\mathcal{C}, n)$
  - 6:  $\mathcal{X}_1^{M_1} \leftarrow \text{XSpectrum}(M_1)$
  - 7:  $\mathcal{Y}_1^{M_1} \leftarrow \text{YSpectrum}(M_1)$
  - 8: **for**  $i \leftarrow 1$  **to**  $n$  **do**
  - 9:    $M_2^i \leftarrow T(0, 0, \psi_i)M_2$
  - 10:    $\mathcal{X}_2^{M_2^i} \leftarrow \text{XSpectrum}(M_2^i)$
  - 11:    $\mathcal{Y}_2^{M_2^i} \leftarrow \text{YSpectrum}(M_2^i)$
  - 12:    $\Delta_x^i \leftarrow \arg \max \text{CrossCorrelation}(\mathcal{X}_1^{M_1}, \mathcal{X}_2^{M_2^i})$
  - 13:    $\Delta_y^i \leftarrow \arg \max \text{CrossCorrelation}(\mathcal{Y}_1^{M_1}, \mathcal{Y}_2^{M_2^i})$
  - 14:    $T_i \leftarrow \Delta_x^i, \Delta_y^i, \psi_i$
  - 15:    $\omega_i \leftarrow (M_1, TM_2^i)$
  - 16: **return**  $T_1 \dots T_n, \omega_1, \dots \omega_n$
- 

The algorithm returns a set of  $n$  transformations  $T_1, \dots, T_n$ , together with the corresponding  $\omega$  value. By inspecting the range of returned  $\omega$  values it is possible to determine whether multiple hypothesis should be tracked or not.

### D. Improvements

As clarified in the introduction, the approach we present in this paper is aimed to support a system where robots exchange and merge their maps during the mission, rather than after the mission, as opposed to our previous contributions. This assumption requires therefore that maps are quickly merged *on the fly*, without imposing to the robots excessively long idle times during the merging stage. Care should therefore be devoted to achieve the utmost speed in the process without sacrificing the quality of the result. The complexity of computing the DHT of a binary image with  $m$  black pixels is  $O(m \cdot n_\theta)$ , while the cost of computing the Hough spectrum is  $O(n_\theta \cdot n_\rho)$ . As the size of the grid used to compute DHT is usually moderate, it turns out that the first of the two terms is the one dominating the complexity. This fact was already observed by Kyriati and colleagues who proposed the *probabilistic Hough transform* in [16]. Starting from the observation that most lines can be detected even without considering all points in the images being processed, they verified that the resulting Hough transform only slightly deteriorates when just a random subset of pixels is considered

during its computation. Not surprisingly, there is a tradeoff between the size of the random subset and the degeneration of the Hough transform, and the optimal value is application dependent. An alternative to this approach consists in picking the subset of points in a deterministic way rather than by random sampling. For example, we may consider only points  $m_1, m_{1+k}, m_{1+2k}$ , and so on. This latter approach will be indicated as *incremental* approach in the following, because its behavior is completely characterized by the increment  $k$ . These two alternatives may significantly speedup the basic method and will be compared in the next section.

A similar idea can be applied to reduce the cost to compute the  $X$  and  $Y$  spectra. An efficient implementation<sup>2</sup> would cost  $O(m)$ , and in order to improve the speed one may avoid considering all points but rather consider only a subset chosen either randomly or in a deterministic way, as described above.

## IV. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of the algorithm presented in this paper we developed an experimental framework based on maps generated from data sets available on public repositories and also collected by the robots available in our research lab. As the gist of our contribution is in merging occupancy grid maps rather than producing maps from scratch, we have used the GMapping algorithm by Grisetti et al. [17], whose implementation is available on the OpenSlam web-site [18]. All the computations took place on a computer equipped with an Intel dual core processor running at 2.14 GHz with 2 Gb of RAM. The machine runs Linux and the code is written in C++. Indoor maps were generated using two Pioneer 3AT robots equipped with a SICK PLS200 range scanner (see figure 3). Since the main goal of these experiments was to produce maps *interesting* from a merging point of view, robots were remotely driven rather than operating autonomously. In this way we were able to map complicated office environments that would otherwise be hard to navigate by a purely autonomous robot. For what concerns maps created from public available data



Fig. 3. The two P3AT used robots right before the starting of the experiment described in subsection IV-A.

<sup>2</sup>such implementation would not use the formulas provided in equations 3 and 4, but would rather consider just the non traversable cells in the grid map, i.e. the  $m$  black pixels in the associated black and white image.

	$T_1$	$T_2$	$T_3$	$T_4$
$\omega$	0.958	0.902	0.900	0.898

TABLE I

THE FOUR  $\omega$  VALUES ASSOCIATED WITH THE TRANSFORMATIONS PRODUCED WHILE MERGING THE TWO MAPS DISPLAYED IN FIGURE 4.

repositories, we used the *ap\_hill* set available in [19]. Such data set has been used also to test other map merging algorithms and is therefore a good benchmark for cross validation and comparisons.

#### A. Merging maps of large environments

The first set of experiments aims to verify the usefulness of the proposed algorithm when merging large maps built by two robots exploring the Science and Engineering building at UC Merced. Figure 4 shows the results of a typical run. The two robots started at the location marked with the letter S in the leftmost panel of figure 4, and they explored different parts of the building. To give a reference about the size of the environment, the reader should consider that the length of the long horizontal corridor displayed in the maps is about 67m. Table I shows the  $\omega$  values associated with the transformations produced by the algorithms (with  $n = 4$ ). Similar results were observed throughout the tests performed.

#### B. Impact of sampling

In order to evaluate the impact of the improvements described in subsection III-D we considered the *ap\_hill* data set available in the *radish* robotics data repository. Such set includes data collected by 4 robots independently exploring an indoor environment. The four partial maps are displayed in figure 5. The impact of randomized or deterministic sampling while computing the Hough transform or the  $X$  and  $Y$  spectra has been evaluated as follows. We considered every value of  $k$  between 1 and 20, and different percentages of points. For each of them we run 100 experiments. In each of them two of the four maps are randomly selected, and a random rotation and translation is applied to the first. The algorithm then computes the best transformation (i.e. the one with the highest associated  $\omega$  value) merging the two. To ensure a fair comparison, the random number generator is always restarted with the same seed, so that in each batch the same set of problems is solved. Figure 6 shows the trend of the computational time and acceptance index for different percentages of considered points. It can be observed that while the percentage of points decreases from 50 down to 5 the average computational time roughly halves (from 360ms to 180ms). At the same time the quality of the result is virtually not affected, as average the  $\omega$  value varies only marginally and the variation is limited. The reader should observe that both maps have a 530 rows and columns. A different behavior is observed for the deterministic sampling, as illustrated in figure 7 for different values of  $k$ . While the time decreases according to the expectations, much higher

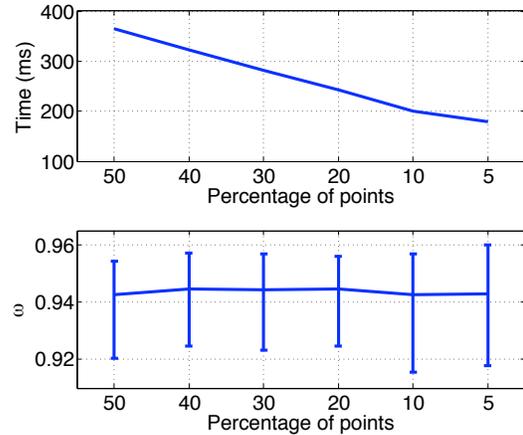


Fig. 6. Impact of random sampling on the Hough transform in term of computational time and result accuracy. The bottom panel displays the average  $\omega$  value, and the associated error bars extend up to the lowest and highest values recorded.

variations for the  $\omega$  value are observed. The randomized approach therefore turns out to be much more robust.

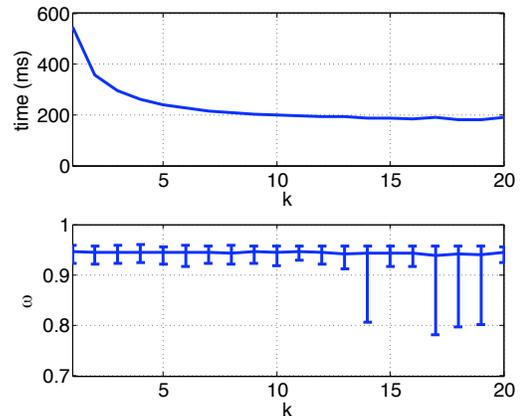


Fig. 7. Impact of deterministic sampling on the Hough transform in term of computational time and result accuracy. The bottom panel displays the average  $\omega$  value, and the associated error bars extend up to the lowest and highest values recorded.

## V. CONCLUSIONS

In this paper we have presented a novel algorithm for merging multiple occupancy grid maps produced by a multi-robot system exploring an indoor environment. The algorithm offers some features not present in related work. More specifically, it never requires that robots localize each other in their respective maps; it does not imply that robots know their relative localization when they start the mission; finally, it produces a set of weighted transformations yielding the desired merging.

The algorithm we described builds upon discrete Hough transform, and the related concept of Hough spectrum. Experimental validation was performed using maps produced from public data repositories and by two mobile robots

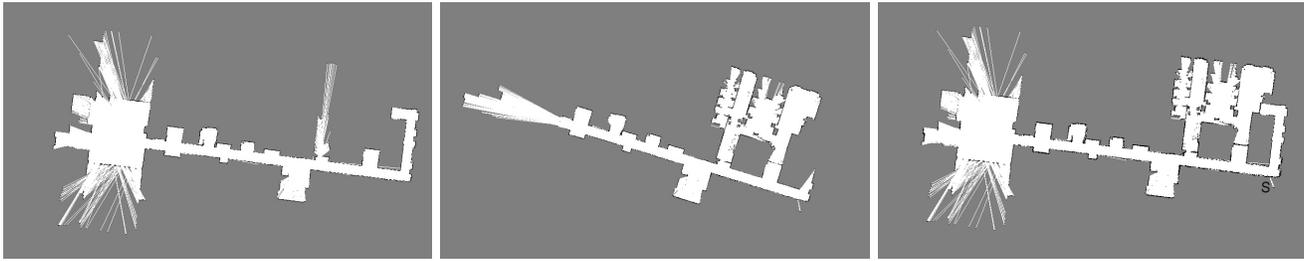


Fig. 4. The left and central panels show the two maps built individually by the two robots. The right one illustrates the map obtained by merging the top ones. The location indicated with the *S* letter is the position where the two robots depicted in figure 3 started their mission.

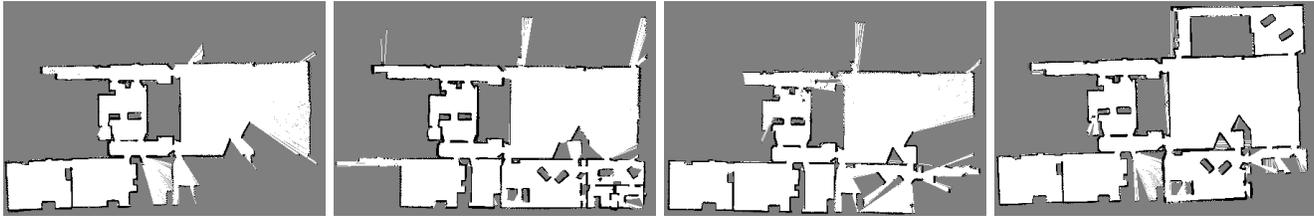


Fig. 5. The four partial maps used to evaluate the impact of different sampling strategies. At each iteration two of the maps are randomly chosen, the first one is randomly rotated and translated, and the best transformation combining them back together is determined by the algorithm.

mapping the engineering building at UC Merced. We have investigated various ways to increase the speed of the discrete Hough transform and experimentally verified that there is a reasonable tradeoff between speed and accuracy of results when subset of points are picked randomly. Deterministic sampling, instead, appears to be less robust. Maps obtained from public available data repositories can be consistently merged in less than 200 ms. Although the results we described are only referred to the case of merging two maps, the idea can be applied to merge an arbitrary number of partial maps by repeatedly applying the algorithm to combine partial results. The technique we propose has two limitations that can be however overcome. First, in order to merge two maps, it is necessary that the two share at least a common part. If this is not the case, the two cannot be fused together. It is important to observe that when this does not happen the algorithm still produces a result, but the associated low  $\omega$  value indicates that the result is to be discarded. Secondly, very long runs may produce maps that are locally consistent, but may be bent and distorted globally. In situations like that a single rigid transformation is not sufficient, and one should rather consider non-rigid transformations allowing different local adjustment. To the best of our knowledge, none of the algorithms for map merging proposed so far properly deals with this problem. A further challenging extension that may be considered in the future concerns merging three dimensional occupancy grids.

#### REFERENCES

- [1] L. Parker, "Current state of the art in distributed autonomous mobile robots," in *Distributed Autonomous Robotic Systems 4*, L. Parker, G. Bekey, and J. Barhen, Eds. Springer, 2000, pp. 3–12.
- [2] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robots*, accepted for publication.
- [3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2006.
- [4] S. Carpin, A. Birk, and V. Jucikas, "On map merging," *Robotics and autonomous systems*, vol. 53, no. 1, pp. 1–14, 2005.
- [5] A. Birk and S. Carpin, "Merging occupancy grids from multiple robots," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.
- [6] A. Howard, L. Parker, and G. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.
- [7] D. Fox, J. Ko, K. Konolige, B. Limketai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, 2006.
- [8] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.
- [9] G. Dedeoglu and G. Sukhatme, "Landmark-based matching algorithm for cooperative mapping by autonomous robots," in *Distributed Autonomous Robotic Systems 4*. Springer, 2000, pp. 251–260.
- [10] S. Williams, G. Dissanyake, and H. Durrant-Whyte, "Towards multi-vehicle simultaneous localisation and mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002, pp. 2743–2748.
- [11] W. Huang and K. Beever, "Topological map merging," *International Journal of Robotics Research*, vol. 24, no. 8, pp. 601–613, 2005.
- [12] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, pp. 977–1000, 2003.
- [13] A. Censi, L. Iocchi, and G. Grisetti, "Scan matching in the Hough domain," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2739–2744.
- [14] R. Duda and P. Hart, "Use of the hough transform to detect lines and curves in the pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [15] T. Röfer, "Using histogram correlation to create consistent laser scan maps," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002, pp. 625–630.
- [16] N. Kiryati, Y. Eldar, and A. Bruckstein, "A probabilistic Hough transform," *Pattern recognition*, vol. 24, no. 4, pp. 303–316, 1991.
- [17] S. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 2432 – 2437.
- [18] (2007) Openslam. <http://www.openslam.org>. [Online]. Available: <http://www.openslam.org>
- [19] Radish, "The robotics data set repository," <http://radish.sourceforge.net>, 2007.