2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Scene Matching between a Map and a Hand Drawn Sketch Using Spatial Relations

Gaurav Parekh, Marjorie Skubic, Ozy Sjahputera, and James M. Keller

Abstract—**The goal of this work is to determine the object correspondence between a sketched map and the scene depicted by the sketch, e.g., as represented by an occupancy grid map (OGM) built by a robot. We describe a novel method based on spatial relations for accomplishing this task. Our method is based on using the histogram of forces as scene descriptors. We generate a correspondence map between two scene descriptors and evaluate its confidence. From this map, we generate a one-to-one object correspondence map for the two scenes such that the object correspondence confidence value is maximized. Challenges include the fact that the two scenes may differ in terms of the shape and size of the objects, their orientation, and the objects might be shifted due to translation. The approach is evaluated using several hand drawn sketches that were collected as a part of a user study. We believe that the ability to perform scene matching will make our sketch interface more robust and easier to use, thereby providing us with a more intuitive way of communicating with the robots.**

Keywords- scene matching, sketch-based navigation, spatial relations, histogram of forces

I. INTRODUCTION

Human robot interaction has been an area of great interest for many AI and robotics researchers. Several strategies have been proposed over the years that would enable us to interact and communicate with robots in the same way we interact and communicate with people. One such strategy has been the use of hand drawn sketch maps that can be used to direct mobile robots along a designated path or to a target location, using an interface which is intuitive to humans.

In our previous work [1][2], we have shown how a sketch can be used as an effective means of communicating with one or more robots. One of the main constraints in [2] has been that it required the user to identify the different objects in the scene and label them correctly on the sketch. If, due to human error, the user did not label the objects correctly then the robots would not be able plan their paths effectively and as a result would be unable to navigate through the scene. Also, it is possible that in the real world the user may not know the correct labels associated with each object. The work proposed in this paper relaxes this constraint and, as a result, makes the sketch interface more robust.

The goal of this work is to determine the object correspondence between a sketched map and the scene depicted by the sketch, e.g., as represented by an occupancy grid map (OGM) built by a robot. We propose a method based on spatial relations for accomplishing this task.

Various methods exist in the literature for matching two images that are seen from different viewing perspectives. In [3], Boland *et al*. used the gradient quantization approach to produce an edge map for scene matching. The matching performance was improved by finding an optimal threshold for gradient quantization. Wong [4] proposed a polynomial estimation approach that estimated the geometric transformation using pairs of corresponding points from both the images as matching elements. Wong also used edge information as matching elements in [5]. Shi *et al*. [6] proposed an algorithm based on Fourier phase correlation and edge enhancement. The algorithm assumed that the considered images had similar edge orientations.

In this paper we present a scene matching strategy that uses spatial relations, i.e., the relative position between objects in a scene. Several methods have been proposed to capture the spatial relations between objects. Gader [7] used mathematical morphology to define spatial relationships, fuzzifying the standard method for computing binary morphology using the extension principle. Krishnapuram et al. [8] proposed the use of angles to define the relative directional position of on object. This concept was used in the angle histogram method introduced by Miyajima and Ralescu [9]. In this paper, we use a generalization of the angle histogram method known as the force histogram method to capture the spatial relations between objects, as proposed by Matsakis and Wendling [10].

Some previous work has been done in the area of using spatial relations for scene matching. Linguistic descriptions of spatial relations were used by Keller et al. in [11] as matching elements in scene matching problems. Sjahputera *et al*. [12] used the force histograms to match two object pairs based on the similarity of their spatial relations and extended this to scene matching in [13]. A better matching algorithm was introduced in [14] that used force histogram transformations to optimize the histogram similarity.

In [15][16][17] and [18], a set of force histograms was used as image descriptors and two methods were introduced for generating a correspondence map between two image descriptors. From this map, an object correspondence

This work was supported in part by the U.S. Naval Research Lab

G. Parekh, M. Skubic and J. M. Keller are with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO. (G. Parekh phone: 480-627-9802; e-mail: gmppx5@mizzou.edu; M. Skubic e-mail: skubicm@missouri.edu; J. M. Keller e-mail: kellerj@missouri.edu)

O. Sjahputera is with the Center for Geospatial Intelligence, University of Missouri, Columbia, MO. (e-mail: sjahputerao@missouri.edu)

confidence matrix was generated, which contained the degree of confidence for relating an object in one image with the corresponding object in the other image. From this matrix, a one-to-one object correspondence map known as the *OMAP* was generated.

In this paper, we describe an application of this approach by performing scene matching between a map that has been drawn to scale and a hand drawn sketch. For our application we have the top down views of the map and the sketch available to us. Hence, we modify the approach presented in [15][16][17] and [18] to exclude information from sensor pose parameters such as cameras' tilt and swing. Challenges include the fact that the two scenes may differ in terms of the shape and size of the objects, their orientation, and the objects might be shifted due to translation. We present an approach that is robust enough to take these challenges into consideration.

In Sec. II we explain our approach and the algorithms associated with it. Sec. III presents experimental results for matching a real map with various sketches (the sketches were collected as a part of a user study). Finally, in Sec. IV, we summarize and discuss future plans for this approach, including making it faster and more robust.

II. METHODOLOGY

We assume that the two scenes have been segmented. Segmentation on the sketchpad is handled by considering each object as a closed polygon [19] [20]. Segmentation of the OGM is accomplished through a series of filtering operations as described in [21]. Labels are applied for convenience but are not used in the scene matching.

The goal is to match each object in the template scene with its corresponding object in the argument scene. In our case, the template scene (S) is the sketch drawn by a user on the sketchpad and the argument scene (S') is the Ovidence Grid Map (OGM). Since the sketch is drawn by a human and the OGM is generated using sensors on the robot; it is very likely that the template and argument scenes differ in scaling, orientation, object size, object shape, and even objects shifted due to translation. To compare the scenes it is important to neutralize these differences.

We can neutralize these differences by generating F-histograms between different pairs of objects and then performing some geometric transformations on the F-histograms of the template scene (sketch) so that it aligns up correctly with the argument scene (OGM). The transformations that neutralize the scaling, orientation, and translational differences can be given as follows [14]:

Scaling: We compute the histograms between different pairs of objects in each scene and then compute their means. We then calculate a scaling factor (ℓ) and apply the transformation:

$$F_r^{a1b1} = \ell^{3-r} * F_r^{ab}$$

where,
F_r^{ab} is a histogram relation between objects a and b in S.
F_r^{a1b1} is the transformed histogram relation.
$r = 0$ for histogram of constant forces.
This takes care of scaling differences between the scenes.

Orientation: We compute the centroids (i.e. main directions) for all the histograms in the template as well as the argument scenes and then apply the transformation:

$$F_r^{a2b2} = F_r^{a1b1}(\theta - \rho).$$

where,
F_r^{a1b1} is the transformed histogram relation from above.
F_r^{a2b2} is the same histogram relation after orientation transformation.
ρ gives the direction in which the histogram must be shifted to account for orientation differences.
θ represents a direction and $\theta \in (0^\circ, 180^\circ)$

Translation: The F-histograms are unaffected by translation of object positions; hence no transformation is required.

Once the effects of scaling, orientation and translation have been neutralized we compute the similarity between different pairs of histograms. There are various similarity measures in the literature [22][23]. We use the Normalized cross-correlation index that is given by:

$$\mu(h_1, h_2) = \frac{\sum_\theta h_1(\theta) * h_2(\theta)}{\sqrt{\sum_\theta h_1^2(\theta)}\sqrt{\sum_\theta h_2^2(\theta)}}$$

where, h_1 is $F_r^{a'b'}$ and h_2 is F_r^{a2b2}; $h_i(\theta)$ represents the histogram value for direction θ.

Finally, we save the 3-tuple output after matching each pair of histograms:

$$\sigma \leftarrow \mu(F_r^{a'b'}, F_r^{a2b2}) \; ; \; R \leftarrow \rho \; ; \; L \leftarrow \ell.$$

We use the F-histogram method to capture the spatial relations information between a pair of objects in a scene. Hence by using a set of F-histograms we can define the spatial relations between all the objects in a scene. Such a set of F-histograms is known as a scene descriptor [18]. It can be given by:

$$F_rD = \{ F_r^{ab} \mid O_aO_b \in \text{S where } a,b = 1,\ldots,N \text{ and } a<b\}$$

$$F_rD' = \{ F_r^{a'b'} \mid O_{a'}O_{b'} \in \text{ S' where } a',b' = 1,\ldots,N' \text{ and } a' \neq b'\}$$

where N = Number of objects in S and
N' = Number of objects in S'

For ease of notation we index the elements of F_rD and F_rD' based on the labels of the objects:

$$F_rD = \{ F_r^{(c)} \mid c = 0,\ldots,Q-1\}$$

where $Q = N*(N-1) / 2$ and

$$c = (b-1) + ((a-1)*N) - \sum_{j=0}^{a-1}(j+1)$$

$$F_rD' = \{ F_r^{(c')} \mid c' = 0,\ldots,Q'-1\}$$

where $Q' = N'*(N'-1)$ and

$$c' = (b'-1) + ((a'-1)*N') - \sum_{j=0}^{a'-1}(j+1)$$

F_rD' contains $F_r^{a'b'}$ and its dual $F_r^{b'a'}$. Its dual is indexed at:

$$c'' = c' + (Q'/2)$$

Also, if $F_r^{(c)} = F_r^{ab}$ then $a = arg(F_r^{(c)})$ and $b = ref(F_r^{(c)})$.

Scenes S and S' are represented by their F-histogram descriptors F_rD and F_rD' respectively. We consider all

possible histogram relations $F_r^{(i)} \rightarrow F_r^{(j')}$ defined by $F_rD \times F_rD'$. Each histogram relation $F_r^{(i)} \rightarrow F_r^{(j')}$ is evaluated using the histogram matching algorithm $\mu(F_r^{(i)}, F_r^{(j')})$ which returns a 3-tuple output (σ, R, L). The normalized 3-tuple output is denoted as $(\hat{\sigma}, \hat{R}, \hat{L})$. The features $\hat{\sigma}$ and $\hat{R}$ are normalized to [0, 1]. The common values for $\hat{L}$ are normalized to [0, 1], even though $\hat{L}$ as a whole, is not bounded to a specific range of values. The features can be normalized using the method below [22][24]:

1. The value of σ is already in [0, 1], therefore $\hat{\sigma} = \sigma$.
2. Since R is periodic we use the following normalization:
$$\hat{R} = \begin{cases} (360^\circ - R)/180^\circ & \text{for } R > 180^\circ \\ R/180^\circ & \text{otherwise.} \end{cases}$$
3. The scaling factor is generally in the range of [$^1/_2$, 2] and it can be normalized using the following equation:
$$\hat{L} = C * ln(L).$$
where C is found to be 2.164.

We can regard the normalized 3-tuple output as a vector $\vec{x}_{ij'} = [\hat{\sigma}\ \hat{R}\ \hat{L}]$. The set of all $\vec{x}_{ij'}$ generated from $F_rD \times F_rD'$ is denoted by X where $|X| = Q*Q'$.

Let $d(\vec{x}_{ij'}, \vec{x}_{pq'})$ be the Euclidean distance between two vectors where $\vec{x}_{ij'}, \vec{x}_{pq'} \in X$, then this distance can be computed as follows:

$$d(\vec{x}_{ij'}, \vec{x}_{pq'}) = [(\hat{\sigma}_{ij'} - \hat{\sigma}_{pq'})^2 + (\hat{R}_{ij'} - \hat{R}_{pq'})^2 + (\hat{L}_{ij'} - \hat{L}_{pq'})^2]^{1/2} \quad (3.1)$$

A. FMAP Generator Algorithm (FMG):

The concept of the *FMAP* is to build the evidence for object correspondence between images S and S'. It was proposed in [17] and reported further in [15], [16], and [18]. If S $\subseteq$ S', S$\rightarrow$S' is defined by a one-to-one *FMAP* from F_rD to F_rD', $F_r^{(i)} \rightarrow F_r^{(j')} \in$ *FMAP*, and *FMAP* $\subseteq F_rD \times F_rD'$. *FMAP* is bijective if S and S' contain the same number of objects. Even though histogram duals are included in F_rD', we do not use both the histogram and its dual in the *FMAP* [17]. This rule and the one-to-one property make up the *FMAP integrity properties*. If the properties are satisfied, a completed *FMAP* is called a legal *FMAP*. The *FMAP* confidence (ζ) is calculated based on the similarity of each recovered parameter $[\hat{\sigma}\ \hat{R}\ \hat{L}]$ across all histogram correspondences in the *FMAP* [15][16][17][18]. If S and S' capture the same objects, then the correct *FMAP* should contain all the correct histogram correspondences $F_r^{(i)} \rightarrow F_r^{(j')}$. According to Matsakis *et al.* [14] under ideal scene conditions (2D, vector data, orthographic projection), these histogram correspondences are expected to produce identical $[\hat{\sigma}\ \hat{R}\ \hat{L}]$, thus $\zeta = 1$. However, if the image is less than ideal then a correct *FMAP* is likely to have $\zeta < 1$, but ζ should be close to 1.

Several *FMAP* generator (FMG) algorithms were proposed to build a legal *FMAP* [17]. These algorithms perform a sequential search in X from a seed point $\vec{y}$ where $\vec{y}$ is a vector whose elements are within the value ranges of the elements in X. The simplest FMG algorithm is the *Nearest Neighbor (NN)* method that builds the *FMAP* by finding Q vectors in X closest to $\vec{y}$ that satisfy the *FMAP* integrity properties. FMG-NN method has been used in scene matching with some success in [15] and [16]. In the NN method, the candidate vectors for the *FMAP* are ranked solely based on their distances from the seed point $\vec{y}$ (a closer vector is a more suitable candidate). Using Euclidean distance, this method works well if the correct $F_r^{(i)} \rightarrow F_r^{(j')}$ are clustered in a tight hypersphere with $\vec{y}$ as the cluster center. With less than ideal scenes and a high variation of object shapes and/or relative position (as is expected in hand-drawn sketches) this condition may not prevail. This was confirmed by our initial results where FMG-NN did not do well in this application.

Recognizing the limitation of FMG-NN while dealing with our application, we used an improved FMG method where the fitness values of candidate vectors are calculated based on their "closeness" to vectors already in the *FMAP* at that point [16][17]. This method is called the *Fuzzy Sequential Nearest Neighbor (FSNN)* method. The degree of "closeness" is assessed using the fuzzy membership function given below:

$$\mu_{close}(d) = \frac{1}{1 + e^{\alpha*(d-m+kv)/v}} \quad (3.2)$$

where,
α controls the steepness of the membership function;
m and v are the average and variance of distances between any pair of vectors $(\vec{x}_{ij'}, \vec{x}_{vz'})$ where $\vec{x}_{ij'}$ and $\vec{x}_{vz'}$ represent two histogram relations that do not conflict with each other. Thus both can exist in the same *FMAP*. They can be computed using following equations:

$$m = \frac{1}{QQ'(Q-1)(Q'-2)} \sum_{i=0}^{Q-1} \sum_{j'=0}^{Q'-1} \sum_{v=0 \neq i}^{Q-1} \sum_{\substack{z'=0 \neq j' \\ z' \neq j'+(Q'/2) \\ z' \neq j'-(Q'/2)}}^{Q'-1} d(x_{ij'}, x_{vz'}) \quad (3.3)$$

$$v = \frac{1}{QQ(Q-1)(Q'-2)} \sum_{i=0}^{Q-1} \sum_{j'=0}^{Q'-1} \sum_{v=0 \neq i}^{Q-1} \sum_{\substack{z'=0 \neq j' \\ z' \neq j'+(Q'/2) \\ z' \neq j'-(Q'/2)}}^{Q'-1} [d(x_{ij'}, x_{vz'}) - m]^2 \quad (3.4)$$

The candidate vector that has the best fitness and is in compliance with the *FMAP* integrity properties is added to *FMAP*. We found that the FSNN performed better than the NN method at matching the sketches with the scene maps. Based on this result, we use the FMG-FSNN to generate our *FMAP*. The results based on this algorithm are discussed in the section III. The algorithm can be written as follows:

1. Let A be the assignment order of vectors in FMAP, *i.e.* ${}^A\vec{z}$ is the A-th vector assigned to FMAP. Initialize A = 1. Let ${}^AFIT$ be the fitness value of ${}^A\vec{z}$.
2. Find the closest vector to $\vec{y}$: $\vec{x}_{IJ'} = \min_{d(\vec{y}, \vec{x}_{ij'})} (\vec{x}_{ij'})$, $\vec{x}_{ij'} \in X$.

3. Initialize FMAP: select $F^{(I)} \rightarrow F^{(J')}$, $FMAP[I] = J'$, ${}^{A}\vec{z} = \vec{x}_{IJ'}$, ${}^{A}FIT = \mu_{close}(d({}^{A}\vec{z}, \vec{y}))$, where $\mu_{close}(d) = \left(1 + e^{\alpha(d-m)/s}\right)^{-1}$, α is the steepness constant, m and s are the mean and standard deviation of $d(\vec{x}_{ij'}, \vec{x}_{pq'}), \vec{x}_{ij'}$ and $\vec{x}_{pq'} \in X$ such that $i \neq p$ and $j' \neq q'$ (FMAP is one-to-one), and j' is not the dual of q', *i.e.* $\vec{x}_{ij'}$ and $\vec{x}_{pq'}$ obey the FMAP integrity properties.
4. **Do Until** FMAP is full ($A = Q$)
   - 4.1 For each $\vec{x}_{ij'} \in X$ satisfying FMAP integrity properties and $\vec{x}_{ij'} \notin$ FMAP, calculate candidate fitness $CFIT_{ij'} = \frac{1}{A}\sum_{a=1}^{A}\left[({}^{a}FIT) \cdot \mu_{close}(d(\vec{x}_{ij'}, {}^{a}\vec{z}))\right]^{1/2}$
   - 4.2 Find the best fitting vector (highest *CFIT*): $\vec{x}_{IJ'} = \max_{CFIT_{ij'}}(\vec{x}_{ij'})$.
   - 4.3 Assign the best fitting vector to FMAP: $FMAP[I] = J'$, ${}^{A+1}\vec{z} = \vec{x}_{IJ'}$, ${}^{A+1}FIT = CFIT_{IJ'}$.
   - 4.4 Increment A.

   End Until

B. Particle Swarm Optimization (PSO).

FMG is a method for finding the optimal *FMAP* given a seed point $\vec{y}$. In order to find a *FMAP* with the highest possible confidence (ζ) we need to use a search algorithm. We use the particle swarm optimization technique for this purpose [18][25]. The parameters we used for the PSO are given below:

(1) Neighborhood: We used a star neighborhood. In a star neighborhood, each particle can communicate with every other particle, forming a fully connected social network.

(2) Search space: 3- dimensional ($\hat{\sigma}, \hat{R}, \hat{L}$).

(3) Initialization: In [15][16][17] and [18] particles are initialized using a selective initialization scheme. This was done to provide the PSO with the likely starting points from where it could 'grow' the *FMAP*. Such a scheme has not been used in this work. In this paper, $\hat{\sigma}$, $\hat{R}$ and $\hat{L}$ are all initialized randomly; $\hat{\sigma}$ and $\hat{R}$ are initialized between [0, 1] and $\hat{L}$ between [-0.5, 0.5]. To ensure that all the particles are initialized independent of one another, system time was passed as the seed to the random number generation algorithm.

(4) Fitness function: The *FMAP* confidence value (ζ) is used as the fitness function.

(5) Velocity Restriction: We have constrained the maximum velocity with which a particle can move in any direction. This is done to ensure that the particles don't 'fly' over good regions in the search space. The velocity is restricted to [-0.2, 0.2].

(6) Convergence: We conclude that the algorithm has converged when any of the following conditions are met:

- The maximum number of iterations is reached.
- *gbest* is relatively constant for several iterations.

(7) Velocity Update: The equation for updating the velocity of each particle can be given by:

$$\vec{v}_k(t) = \vec{v}_k(t\text{-}1) + \rho_1(\vec{x}_{pbest} - \vec{x}_k(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_k(t)) \quad (3.5)$$

where, $\rho_1 = r_1 c_1$; $\rho_2 = r_2 c_2$;
$r_1, r_2 \sim U(0,1)$; $c_1 + c_2 \leq 4$
t is the iteration number.
k is the particle index.

(8) Position Update: The position of a particle can be updated using the following equation:

$$\vec{x}_k(t) = \vec{x}_k(t\text{-}1) + \vec{v}_k(t) \quad (3.6)$$

The algorithm is given below:

1 Set $t = 0$.
2 Initialize K particles position $\vec{x}_k(t)$ randomly.
3 Initialize $gbest = 0$.
4 **Do Until** one of the convergence criteria is met
- 4.1 **For** each particle **Do**
  - 4.1.1 Get $FMAP$ = FMG ($\vec{x}_k(t)$).
  - 4.1.2 Calculate ζ from *FMAP*.
  - 4.1.3 **If** $\zeta > pbest$ **Then**
    - 4.1.3.1 $pbest = \zeta$, $\vec{x}_{k,pbest} = \vec{x}_k(t)$.

    End For (4.1.3)
  - 4.1.4 **If** $\zeta > gbest$ **Then**
    - 4.1.3.1 $gbest = \zeta$, $\vec{x}_{gbest} = \vec{x}_k(t)$.

    End If (4.1.4)
- 4.2 **For** each particle **Do**
  - 4.2.1 Update $\vec{v}_k$ using (3.5).
  - 4.2.2 Update $\vec{x}_k$ using (3.6).

  End For (4.3)
- 4.3 $t \leftarrow t + 1$

End Until (4)

The search algorithm helps us find the best possible *FMAP* with the maximum possible confidence (ζ). We now wish to translate this *FMAP* into a set of object correspondences that allow a human observer to easily identify the mapping / registration of objects across the two scenes. For this purpose, an Object Correspondence Confidence Matrix (*OCCM*) is developed from which we construct a one-to-one object correspondence map known as the *OMAP*.

C. Object Correspondence Confidence Matrix (OCCM).

In the FMAP, each histogram correspondence $F_r^{(i)} \rightarrow F_r^{(j')}$ implies an object correspondence $(O_a, O_b) \rightarrow (O_{c'}, O_{d'})$ where, (O_a, O_b) represents an object pair in S such that $a<b$; and $(O_{c'}, O_{d'})$ represents an object pair in S' such that $c' \neq d'$. To determine the one-to-one object correspondence, an $N \times N'$ matrix is constructed, where N and N' are the number of objects in S and S' respectively. This matrix is known as the Object Correspondence Confidence Matrix (*OCCM*) [17][18]. Each element of this matrix represents the confidence for a certain object correspondence. For example, an element $OC_{gh'}$ represents the confidence with which object O_g in S corresponds to object $O_{h'}$ in S' $(O_g \rightarrow O_{h'})$.

Now, the correspondence $(O_a, O_b) \rightarrow (O_{c'}, O_{d'})$, gives rise to two sets of object correspondences: $\{O_a \rightarrow O_{c'}, O_b \rightarrow O_{d'}\}$ and $\{O_a \rightarrow O_{d'}, O_b \rightarrow O_{c'}\}$. Both sets must be taken into account to arrive at the correct object correspondence. The histogram correspondence, $F_r^{(i)} \rightarrow F_r^{(j')}$ provides some evidence to support the four different object correspondences: $O_a \rightarrow O_{c'}$, $O_b \rightarrow O_{d'}$, $O_a \rightarrow O_{d'}$, and $O_b \rightarrow O_{c'}$. The measure of confidence for $F_r^{(i)} \rightarrow F_r^{(j')}$ is given by $\sigma_{ij'}$. Therefore, $\sigma_{ij'}$ can be distributed as 'partial' confidence to the four object correspondence confidence values:

$$OC_{ac} = OC_{ac} + \sigma_{ij'} \qquad OC_{bc} = OC_{bc} + \sigma_{ij'}$$
$$OC_{ad} = OC_{ad} + \sigma_{ij'} \qquad OC_{bd} = OC_{bd} + \sigma_{ij'}$$

The same process is repeated for each histogram correspondence in *FMAP*. The *OCCM* accumulates the partial confidence supplied by each histogram correspondence.

D. One-to-one Object Map (OMAP).

Upon obtaining the completed *OCCM* matrix, we need to determine the one-to-one object correspondence map (*OMAP*) such that the object correspondence confidence value is maximized [17][18]. To do this, the *OMAP* is initialized as an empty 1-D array with *N* elements. The array is then filled with object relations that had the highest confidence in the *OCCM*. The most confident object relation is added to the *OMAP*. For each subsequent object relation (i.e., $O_g \rightarrow O_{h'}$) two conditions are checked:

(1) *OMAP*[g] is empty.

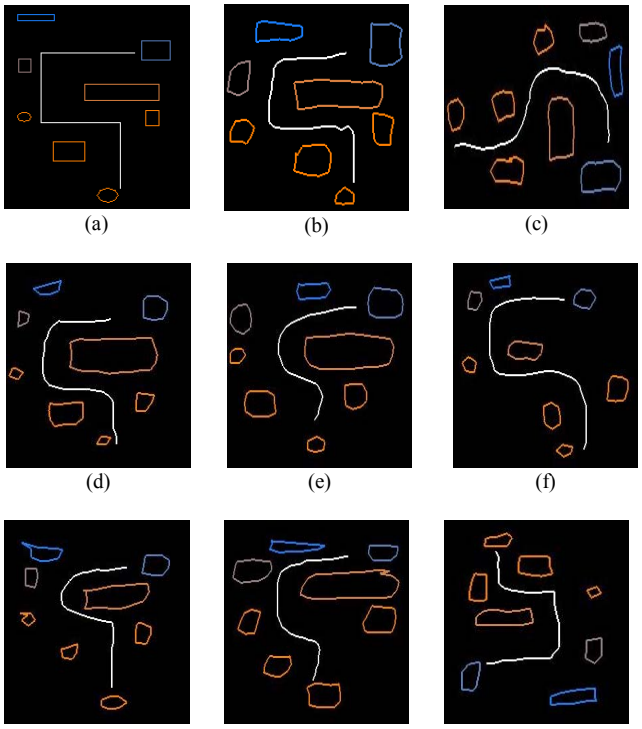
(2) *OMAP*[m] ≠ h', where m = 1,…, *N*

These conditions ensure that the one-to-one property of the *OMAP* is preserved. If the conditions are met then the object relation $O_g \rightarrow O_{h'}$ can be added to the *OMAP* and *OMAP* confidence (Ω) is incremented by $OC_{gh'}$. The *OMAP* thus obtained allows us to easily identify the correspondence of objects between the template (sketch) and the argument (OGM) scenes.

III. Experiments And Results

To test the effectiveness and robustness of our approach we have tried to match several hand drawn sketches with a map of the environment that has been drawn to scale. The sketches were drawn as a part of a user study in which volunteers were presented with a physical scene and then asked to sketch the scene on a PDA sketchpad [19][20]. We expect to get similar results if the map of the environment is replaced by a segmented OGM built using the method described in [21].

Figure 1(a) shows the map of the environment, Figures 1(b)–(i) show the sketches that are used to test the approach. Note that the sketched paths shown in the figures were not used in the scene matching. The results from the experiments are summarized in Table 1.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 1. (a) Physical map, (b) – (i) are sketches used for testing.

TABLE I. SUMMARY OF RESULTS

Sketch	FMAP Confidence (ζ)	OMAP Confidence (Ω)	Correct matches
Case (a)	1.00	1.00	8
Case (b)	0.91	0.72	8
Case (c)	0.88	0.53	8
Case (d)	0.91	0.75	8
Case (e)	0.87	0.57	8
Case (f)	0.92	0.69	8
Case (g)	0.92	0.74	8
Case (h)	0.86	0.52	6
Case (i)	0.88	0.66	8

From the table we can conclude that our approach has given good results on all but one of the sketches. Case (a) is an experiment in which scene matching was carried out using the map of the environment as the template scene as well as the argument scene. It represents an ideal scenario and hence it is not a surprise that we get an ideal result. Cases (b) – (i) are experiments when scene matching was carried out between the map and sketches (b) – (i) respectively. The approach found an *FMAP* with a high confidence value in all cases. This *FMAP* turned out to be the correct *FMAP* in all but one of the cases and as a result we got an object mapping that had a high confidence. Case (h) represents an experiment in which we could not get a perfect object mapping. To investigate this case we arranged the objects such that the correct *FMAP* would be known to us and then we evaluated the fitness of this *FMAP*. We found its fitness (ζ) to be 0.882 which is clearly higher than

the best fitness that the swarm could find. Thus we concluded that, if instead of the PSO, we had used an exhaustive search then we would have found the correct *FMAP*. Of course, the trade-off here is between the speed and cost of computation versus the accuracy.

We are currently investigating different approaches for initializing the swarm in PSO so that the search converges faster and we avoid the problem of getting stuck in a local maxima.

IV. Concluding Remarks

We have presented a technique to perform scene matching between a map of the environment and a sketch using spatial relations. This work was done with the motivation of making the robot sketch interface more robust and easier to use. Considering the application, certain modifications were made to the algorithm described in [18]. Features such as tilt and swing angle of the camera were excluded. Also, the selective initialization scheme used in the previous work yielded good results only when the number of objects was restricted to 5 or below. With 8 objects the search space was much more complex. Hence, while searching for the best histogram map (*FMAP*), the particles in the PSO algorithm were initialized randomly thus enabling a wider and more effective coverage of search space.

In the future, we intend to improve on the *FMAP* generator algorithm and the method of computing its fitness. We would also like to make this approach more robust by considering the cases where the two scenes have different numbers of objects. In addition, the current algorithm is computationally intensive and as a result has a drawback of being quite slow. To run this algorithm on a robot we need to speed up the search without affecting its accuracy. Our future efforts will be directed towards this goal.

References

[1] G. Chronis and M. Skubic, "Robot Navigation Using Qualitative Landmark States from Sketched Route Maps," in *Proc. of the IEEE 2004 Intl. Conf. on Robotics and Automation*, New Orleans, LA, April, 2004, pp. 1530-1535.

[2] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz, "Using a Hand-Drawn Sketch to Control a Team of Robots," *Autonomous Robots*, in press.

[3] J. S. Boland, H. S. Ranganath, W. W. Malcolm, "Improved Method for Scene Matching," *IEEE Trans. On Automatic Control,* vol. AC-25. no. 3, June 1980.

[4] R. Y. Wong, "Sensor Transformations," *IEEE Trans. On Systems, Man, and Cybernetics,* vol. SMC-7, no. 12, December 1977.

[5] R. Y. Wong, "Sequential Scene Matching Using Edge Features," *IEEE Trans. On Aerospace and Electronic Systems,* vol. AES-14, no. 1, January 1978.

[6] D. Shi. L. Han, and Y. Liu, "A scene matching algorithm based on the knowledge of object edges," *Proc. IEEE Int. Con5 on Intelligent Processing Systems,* pp. 1442-1445, 1997.

[7] P. D. Gader, "Fuzzy Spatial Relations Based on Fuzzy Morphology", *Proceedings, IEEE Int. Con5 on Fuzzy Systems,* Barcelona, vol. 2, pp. 1179-1183, 1997.

[8] R. Krishnapuram, J. M. Keller, Y. Ma, "Quantitative analysis of properties and spatial relations of fuzzy image regions," *IEEE Trans. Fuzzy Systems,* vol. 1, no. 3, pp. 222-233, 1993.

[9] K. Miyajima, A. Ralescu, "Spatial organization in 2-D segmented images: representation and recognition of primitive spatial relations," *Fuuy Sets and Systems,* vol. *65,* no 23, pp. 225-236, 1994.

[10] P. Matsakis, L. Wendling, "A new way to represent the relative position between aerial objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 634-643, 1999.

[11] J. M. Keller, P. D. Gader, X. Wang, "LADAR scene description using fuzzy morphology and rules," *IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications,* 1999 (CVBVS '99), Proceedings, page(s): 120-129.

[12] O. Sjahputera, J. M. Keller, P. Matsakis, J. Marjamaa, "Histogram based scene matching measures," *Proc. 19th Int'l Conf. North American Fuzzy Information Processing Society*, pp. 392-396, 2000.

[13] O. Sjahputera, J. M. Keller, P. Matsakis, "Scene Matching by Spatial Relationships," *Proc. 22nd Int'l Conf. North American Fuzzy Information Processing Society*, pp. 149-154, 2003.

[14] P. Matsakis, J. M. Keller, J. Marjamaa, O. Sjahputera, "The use of Force Histograms for Affine Invariant Relative Position Description," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 1-18, 2004.

[15] O. Sjahputera and J. M. Keller, J., "Possibilistic c-means in scene matching", *Proceedings, Fourth International Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT)*, 669–675, Sept. 2005.

[16] O. Sjahputera and J. M. Keller, "Scene matching using Fhistogram-based features with possibilistic C-means optimization," *Fuzzy Sets and Systems*, *Special Issue on Image Processing*, revised, 2006.

[17] O. Sjahputera, "Object registration in Scene Matching based on Spatial relationships," Ph.D. Dissertation, Computer Engineering and Computer Science, University of Missouri, Columbia, MO, July, 2004.

[18] O. Sjahputera and J. M. Keller, "Particle swarm over scene matching," *Proceedings, IEEE Swarm Intelligence Symposium (SIS)*, 108–115, June 8-10, 2005.

[19] C. Bailey, "A sketch interface for understanding hand-drawn route maps," M.S. Thesis, Computer Engineering and Computer Science, University of Missouri, Columbia, MO, Dec., 2003.

[20] M. Skubic, C. Bailey and G. Chronis, "A Sketch Interface for Mobile Robots," in *Proc. of the IEEE 2003 Conf. on SMC* , Washington, D.C., Oct., 2003, pp. 918-924.

[21] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial Language for Human-Robot Dialogs," *IEEE Transactions on SMC Part C, Special Issue on Human-Robot Interaction*, vol. 34, no. 2, May, 2004, pp. 154-167.

[22] X. Wang, B. De Baets and E. Kerre, "A Comparitive Study of Similarity Measures," *Fuzzy Sets and Systems*, vol. 73, no. 2, pp. 259-268, 1995.

[23] S. Santini and R. Jain, "Similarity Measures," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871-883, 1999.

[24] B. S. Reddy, B. N. Chatterji, "An FFT based technique for translation. Rotation, and scale-invarient image registration," *IEEE Trans. on Image Proc.,* vol 5, no. 8, pp. 1266-1271, Aug, 1996.

[25] J. Kennedy, R. Eberhart, "Particle Swarm Optimization," *IEEE Int'l Conf. On Neural Networks*, vol. 4, pp. 1942-1948, Nov, 1995.