SCENE MATCHING BETWEEN A QUANTITATIVE MAP AND A QUALITATIVE HAND DRAWN SKETCH

A Thesis
presented to
the Faculty of the Graduate School
at the University of Missouri-Columbia

In Partial Fulfillment
of the Requirements for the Degree

Master of Science

---

by
GAURAV PAREKH

Dr. Marjorie Skubic, Thesis Supervisor

DECEMBER 2007

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

SCENE MATCHING BETWEEN A QUANTITATIVE MAP AND A QUALITATIVE HAND DRAWN SKETCH

presented by Gaurav Parekh

a candidate for the degree of Master of Science

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Marjorie Skubic

Professor James Keller

Professor Kannapan Palaniappan

ACKNOWLEDGEMENTS

This work could not have taken place without the kind assistance and mentoring of my advisor, Dr. Marjorie Skubic. I am grateful to her for introducing me to this unique and challenging project.

Committee members Dr. James Keller and Dr. Kannapan Palaniappan are not without recognition. I am most grateful to them for their time and willingness to review this work and provide expert insightfulness.

This work was generously funded by the Naval Research Laboratories grant number N00173-06-1-G029.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Scene Matching Between a Quantitative Map and a Qualitative Hand Drawn Sketch

Gaurav Parekh

Dr. Marjorie Skubic, Thesis supervisor

ABSTRACT

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. Earlier work [Skubic et al., 2006] has shown that a sketch can be used as an effective means of communicating with one or more robots. One of the main constraints in this work 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 correctly through the scene. The work proposed in this thesis 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 a precise map that has been drawn to scale or an occupancy grid map (OGM) built by a robot. In this thesis, a novel method is proposed for accomplishing this task. Our approach known as the Evolutionary Algorithm for Scene Matching (EASM) is based on spatial relations and an evolutionary algorithm for accomplishing this task. The spatial relations between different objects in a scene are captured using the histogram of forces (F-histogram) method and then an evolutionary algorithm is used as a guided search to find the best histogram relational map. The fact that the two scenes may differ in terms of orientation, object translation and shape and size of the objects has been taken in account. Furthermore, the intended use of this algorithm is in the field of human robot interaction; hence the time taken by the algorithm to converge is of critical importance. Experiments were run using two different scenes and several hand drawn sketches of each scene, which were collected as part of a user study. Scene matching was performed between a quantitative map that was drawn to scale and these qualitative sketches. Scene matching was also performed between occupancy grid maps (OGMs) that were built by a robot and the sketches. A comparison was made between the proposed approach and the Fuzzy Sequential Nearest Neighbor (FSNN) approach that was used in earlier work [Sjahputera, 2004] [Parekh et. al., 2007]. Experiments were also run for cases where the sketches drawn by the user had some missing objects or extra objects. For cases when the sketch and the quantitative map had equal number of objects the proposed

method always found the correct object mapping and converged 124.51 times faster than the FSNN method for a scene with 8 objects. The method also gave very promising results for cases with unequal number of objects and gave a correct mapping in 88.5%of the cases.

1 INTRODUCTION

1.1 Problem Statement

Object mapping/registration is an area of great interest for several researchers across the globe, as it has applications in many fields including computer vision, medical imaging, robotics, etc. In our case, it has an application in the field of human robot interaction. 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 this thesis, an effort has been made to carry out object mapping/registration across two scenes. The scenes that we are trying to match are not regular images; one of them is a hand drawn sketch map and the other is some kind of a quantitative representation of the scene i.e. either an occupancy grid map (OGM) built by a robot or a physical map that has been drawn to scale.

1.2 Overview

The underlying motivation behind this work is to improve on the current robot sketch interface and thereby provide a better and more effective tool for the humans to communicate with the robot. In previous work [Chronis and Skubic, 2004][Skubic et. al., 2006] it was shown that a sketch could be used as an effective means of communicating with one or more robots. However, one of the main constraints in [Skubic et. al., 2006] had 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 correctly through the scene. The work proposed in this thesis relaxes this constraint and, as a result, makes the sketch interface more robust.

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') can be some other kind of a representation of the scene i.e. either an Occupancy Grid Map (OGM) built by a robot or a physical map that has been drawn to scale. To obtain a one-to-one object correspondence between the two scenes the following steps are followed:

1. The first step is to segment the two scenes. Segmentation on the sketchpad is handled by considering each object as a closed polygon [Bailey, 2003] [Skubic et. al., 2003]. Segmentation of the OGM or the physical map is accomplished through a series of image processing operations as described in chapter 4. Labels are applied for convenience but are not used during scene matching.
2. Since the template scene (S) is drawn by a human and the argument scene (S') is generated using sensors on the robot or some other technique; it is very likely that the two scenes differ in size and shape of the objects, the scene orientation, and the objects may even be shifted due to translation. So the second step is to neutralize these differences. The neutralization is

carried out by generating force histograms between different object pairs in a scene and then performing some geometrical transformations on these histograms [Matsakis et. al., 2004]. A brief description of the method used to achieve this is explained later.

3. The force histograms generated in step 2 capture the spatial relationships between an object pair. Hence by using a set of force histograms we can capture the spatial relations between all the objects in a scene. Such a set of histograms is known as a 'scene descriptor'. There are two scene descriptors F_rD and F_rD' that correspond to the scenes S and S' respectively. The third step is to form these scene descriptors [Sjahputera, 2004].
4. The next step is to find a one-to-one histogram relation map known as the FMAP. The FMAP relates a histogram in F_rD to a histogram in F_rD'. Since there are several possible histogram mappings we associate an FMAP confidence value (ζ) with each FMAP. This value is high if all the histograms in F_rD have been related to the correct histograms in F_rD'. The number of possible FMAPs can be given by [Sjahputera, 2004]:

$$M = \left(\frac{N(N-1)}{2}!\right) \times 2^{\left(\frac{N(N-1)}{2}\right)}$$

where, N is the number of objects in a scene. In our experiments there are 8 objects in scene A and 7 objects in scene B. For 8 objects, $M = 8.18 * 10^{37}$. It is obvious that an exhaustive search to find the best FMAP would be too costly. At the same time, it is critical to find the FMAP with the highest confidence (ζ) to arrive at the correct one-to-one object mapping.

5. The final step in this approach is to translate the FMAP into a one-to-one object correspondence map known as the OMAP. The OMAP lets the human observer easily identify the mapping/registration of objects across the scenes. Considering the complexity of the search space, it is possible that the FMAP obtained in step 4 is really good but not perfect. Hence we use an OMAP generation algorithm that tries to build the OMAP as robustly as possible from the given FMAP [Sjahputera, 2004][Sjahputera and Keller, 2005].

The success of this approach greatly depends on the performance of the FMAP search algorithm. To carry out this search, three different approaches were tried, viz., the Nearest Neighbor (NN) approach, the Fuzzy Sequential Nearest Neighbor (FSNN) approach, and the proposed Evolutionary Algorithm for Scene Matching (EASM) approach. The NN and FSNN methods were first proposed in [Sjahputera, 2004] and later reported in [Sjahputera and Keller, 2005] and [Sjahputera and Keller, 2005]. Descriptions of the NN, FSNN and EASM methods are given in chapter 3. In chapter 4, the results are reported for performing scene matching between a precise physical map and a sketch. Scene matching was also performed between an occupancy grid map (OGM) and a sketch. Towards the end of chapter 4 some unique cases are considered where the number of objects in the sketch is different than the number of objects in the quantitative map and the results are reported. Finally the results are summarized in chapter 5 and future directions are discussed.

2 LITERATURE REVIEW

2.1 Sketch Interface

In recent years researchers have shown that sketching can be used as a means for communicating with one or more robots. It enables a novice user who has no experience in robotics to effectively direct the robot to accomplish certain tasks. Hence, sketches prove to be an intuitive tool for human robot interaction. However, using hand drawn sketches present several unique problems. In [Mahoney et. al., 2002] three problems were identified while working with sketches: variability of hand sketches, interactive performance, and extensibility to new configurations. In their work, sloppy drawings lead to what they called failure of co-termination; the stroked marks overshot or undershot in a multi-stroke sketch. This made it difficult to segment the sketch into the proper component parts for recognition and lead to segmentation ambiguity.

Several previous sketch-based systems have been proposed, which rely on a quantitative map. Perzanowski et. al., [Perzanowski et. al., 2001] developed a multi-modal robot interface that had capabilities such as natural language understanding and gesture recognition. In addition, their interface was also capable of accepting inputs from a Palm Pilot or other PDA. They displayed a quantitative map of the environment as the robot traveled through it. To facilitate the communication between the human and the robot they allowed the user to utilize any combination of speech, gesture and PDA inputs in order to accomplish the given task. Lundberg et al. [Lundberg et al., 2003] developed a similar PDA interface for field robots. Their user interface received commands from the user

and returned information about the robot's state, position, and the produced map. The user interface converted the commands received from the user to standardized digital information that was transferred to the robot via a wireless link. Most of the information processing was done onboard the robot. Fong's PDA interface [Fong, Thorpe, and Glass, 2003] included the ability to sketch waypoints on top of a robot-sensed image; this allowed live imagery to be used in the control. Another version of his interface supported the control of multiple robots using human-robot dialogue and the PDA interface [Fong, Thorpe, and Baur, 2003]. The PDA was used to provide a variety of command modes including rate/position control, image based waypoints and map based waypoints.

Kawamura et al. [Kawamura et al., 2003] developed a PDA interface in which a user could specify a robot path by selecting intermediate points on a sketch of the environment. Colored cones were placed in the scene and on the sketch to act as artificial landmarks. The interface had a drag and drop feature to place the objects in a scene. A more robust interface was developed using a PDA in [Skubic et al., 2003] [Bailey, 2003]. The feature that made this interface robust was that it assumed that sketches drawn by the human users would not be very precise and hence only processed qualitative route information and not quantitative information. No restrictions were placed on the number of objects that could be inserted in the scene, no predetermined landmarks were required, there was no restriction on the orientation, and the users were free to sketch the environment as they viewed it. Users of the interface were also asked to evaluate

the usability of the interface and results for this evaluation were reported. In [Chronis and Skubic, 2004] the approach was extended to perform reliably in complex environments. Significant improvements were proposed for achieving better navigation from the extracted landmarks. Spatial relations between qualitative landmarks were extracted from the sketched route and then used by the robot to navigate in the real environment. Then in [Skubic et al., 2006] and [Skubic et al., 2007] this approach was further developed by upgrading the interface from a PDA to a tablet PC. The tablet PC opened a two way communication link between the user and the robot. Now the user did not use the interface as a simple sketch pad, but the user could also see where the robot was in the environment and monitor its sensor readings on the tablet PC. Other new features were also added to the interface that improved its usability. Another significant change was that instead of a single robot, a team of three robots was controlled using the sketch interface.

2.2 Representations for Spatial Relations

Spatial relations are used to describe the relationships between different objects in a scene. In [Winston, 1975], Winston quantified spatial relations to create a program that could learn to recognize line drawing representations of structures by building an abstract representation of a given line drawing and examining the applicability of various internalized structure descriptions. These descriptions were crisp and context sensitive by nature. Rosenfeld in [Rosenfeld, 1979] and [Rosenfeld, 1983] used fuzzy relations to define the properties of

connectedness between two objects. He also defined the properties for adjacency between image objects [Rosenfeld, 1974].

The relative position of an image object with respect to another object was assessed using directions such as RIGHT, LEFT, ABOVE and BELOW in [Keller and Szatendra, 1990]. In [Dutta, 1991], positional constraints and motion constraints were used to represent the spatial reasoning problems. Each type of constraint contained some metric and topological information. This information was modeled using fuzzy sets.

Krishnapuram et al. [Krishnapuram et al., 1993] proposed an aggregation method for capturing the properties and spatial relationships of object regions in a digital image. Each object in the image was represented as a fuzzy region, and spatial relations between fuzzy regions were defined as possibility distributions whose membership functions were defined over α-cut sets of the fuzzy regions. Miyajima and Ralescu [Miyajima and Ralescu, 1994] proposed a histogram based method for determining the spatial relations between two objects in an image. Their method known as histogram of angles was restricted to raster data. They considered all pairs of elements (a, b) where a is an element of object A and b is an element of object B. The angle defined by the pair (a, b) was computed and a counter was setup to tally the number of element pairs (a, b) that define a specific angle. A series of such counters that covered a finite number of directions in [0, 2π] range formed the histogram of angles. A comparative study was made between various methods for defining spatial relations in [Keller and Wang, 1995]. They compared methods like aggregation

method, centroid method and compatibility method. They also proposed a neural network based approach for defining spatial relations in [Wang and Keller, 1999].

Mathematical morphology was used to compute fuzzy spatial relations by Gader in [Gader, 1997]. He also compared his results to the results obtained using the aggregation method proposed in [Krishnapuram et al., 1993].

2.2.1 Histogram of forces

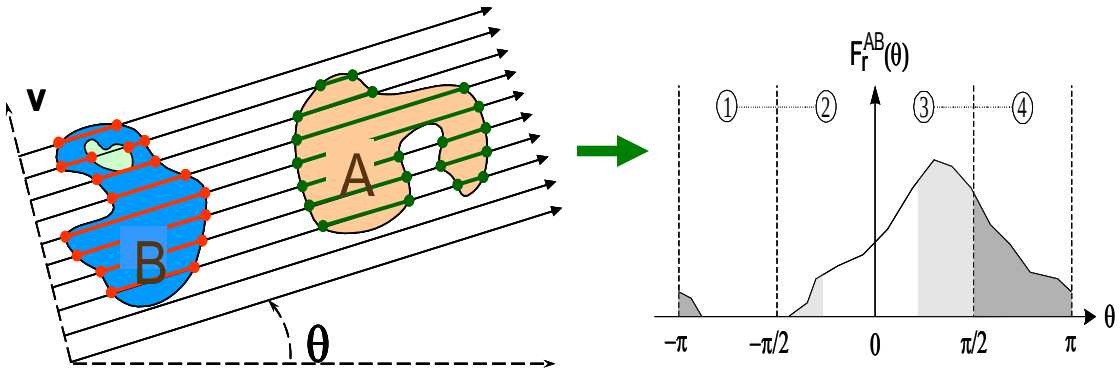


Figure 2.1. Generating force histograms [Matsakis and Wendling, 1999].

The idea of histogram of angles was generalized by Matsakis and Wendling when they introduced the histogram of forces in [Matsakis and Wendling, 1999]. Matsakis and Wendling introduced two different types of histogram of forces viz. histogram of gravitational forces (gives more importance to closer parts of objects) and histogram of constant forces (conceptually similar to histogram of angles). The histogram of forces can be computed using either raster or vector data. To compute the histogram of forces using vector data, the objects were portioned into trapezoidal regions by drawing lines through the object vertices at an angle θ. This process was repeated for each angle θ in the set {0, 2π/d,

4π/d, …, 2(d-1)π/d} for a chosen number of d directions, typically 180 directions. For every set of trapezoids that were formed, the scalar resultant Γ^{IJ} was computed and then the histogram value at θ was updated. Matsakis and Wendling also claimed that this algorithm is very fast and computationally less expensive than the histogram of angles method.

In this work, the histogram of constant forces has been used to assess the spatial relations between different object pairs in a scene.

2.3 Scene Matching

In any scene matching problem, the objective is to associate a region in one scene with the corresponding region in another scene. In this work, there is only one scene but it is captured using two very different styles. One of them is a qualitative representation of the scene (sketch) and the other one is a quantitative representation of the scene (physical map/OGM). Several factors come into play while performing this kind of scene matching, such as the shape and size of the objects, the scene orientation and the translation of the objects.

Scene matching is actually a part of the broader problem of image registration. Based on the matching elements used, scene matching methods can be loosely divided in to the following categories:

1. Algorithms that use image pixel values directly. In [Aschwanden and Guggenbühl, 1992] a comparative study was made between different correlation based registration algorithms that used pixel values directly and the experimental results were presented.

2. Algorithms that transform the image into the frequency domain. Such methods have been used in [Reddy and Chatterji, 1996] [Lin et al., 2001]. In [Reddy and Chatterji, 1996] a FFT based technique was proposed to achieve image registration. This technique was translation, rotation and scale invariant. The FFT based technique was a digital implementation of the Fourier-Mellin transform. The same transform has also been effectively used by [Lin et al., 2001] for digital watermarking of images.
3. Algorithms that use low level features such as points, edges, etc. as matching elements [Wong, 1978] [Boland et al., 1980] [Han and Park, 2000]. Wong proposed a method to perform scene matching between an optical and a radar image by using edge information [Wong, 1978]. He developed a technique for edge extraction and a sequential, hierarchical search technique that was used to perform the matching such that the cross-correlation was maximized. [Boland et al., 1980] introduced a method to perform scene matching between images that had been captured using dissimilar sensors. They used a mean and standard deviation quantization method to set the quantization threshold. This method produced an improved edge map. The edge map was used by a discrete correlation algorithm to perform the matching. The correlation algorithm proposed by them greatly improved the correlation stability. Epipolar geometry was used in [Han and Park, 2000] for contour matching in images. Epipolar geometry is a method that enables us to determine the spatial correlation between points in different images. However, these

images should represent the same scene. In their work, a movable camera was used to capture the same scene from different perspectives.

4. Algorithms that use high level features such as parts of an object, spatial relations between objects, etc. as matching elements. Fuzzy spatial relations were used by [Keller, Gader, Wang, 1999] to produce automatic linguistic descriptions of digital images that were captured using LADAR. A method was also proposed to match a scene with the given linguistic description using a fuzzy rule base. [Sjahputera et al., 2000] performed matching between a scene and the linguistic descriptions used to describe that scene. They used the histogram of forces method to capture the spatial relations between different objects. This approach was extended to scene matching in [Sjahputera et al., 2003]. In this work, they performed scene matching between two different images of the same scene that were captured from different viewing perspectives. The estimated sensor pose parameters and the histogram similarity measure were used as features to perform the matching. The effect of affine transformations on force histograms was studied in [Matsakis et al., 2004] and geometric transformations were proposed that made the F-histograms affine invariant. The use of these transformations in the context of scene matching was studied. In [Sjahputera and Keller, 2005] and [Sjahputera and Keller, 2006], a set of force histograms was used as an image descriptor and two methods were introduced viz. the ‘Nearest Neighbor’ (NN) method and the ‘Fuzzy Sequential Nearest Neighbor’ (FSNN)

method, for generating a correspondence map between the two image descriptors. From this map they built a one-to-one object correspondence map that associated an object in one scene with the corresponding object in the other scene. The Possibilistic C-Means (PCM) algorithm was used to search for best correspondence map between the two image descriptors. In [Sjahputera and Keller, 2005] they also investigated the use of Particle Swarm Optimization (PSO) algorithm to search for the best histogram correspondence map between the two image descriptors.

The work proposed in this thesis falls under the fourth category. Here, a novel method is proposed to perform scene matching, which is called the Evolutionary Algorithm for Scene Matching (EASM) method. It uses spatial relations and an evolutionary algorithm for performing scene matching. The EASM method also builds a one-to-one force histogram correspondence map but in a fundamentally different way. The EASM algorithm has been compared with the FSNN algorithm in a set of experiments described in chapter 4.

3 METHODOLOGY FOR SCENE MATCHING

3.1 Overview of the Scene Matching Method

The entire method for performing scene matching can be summarized using the flow diagram shown below:

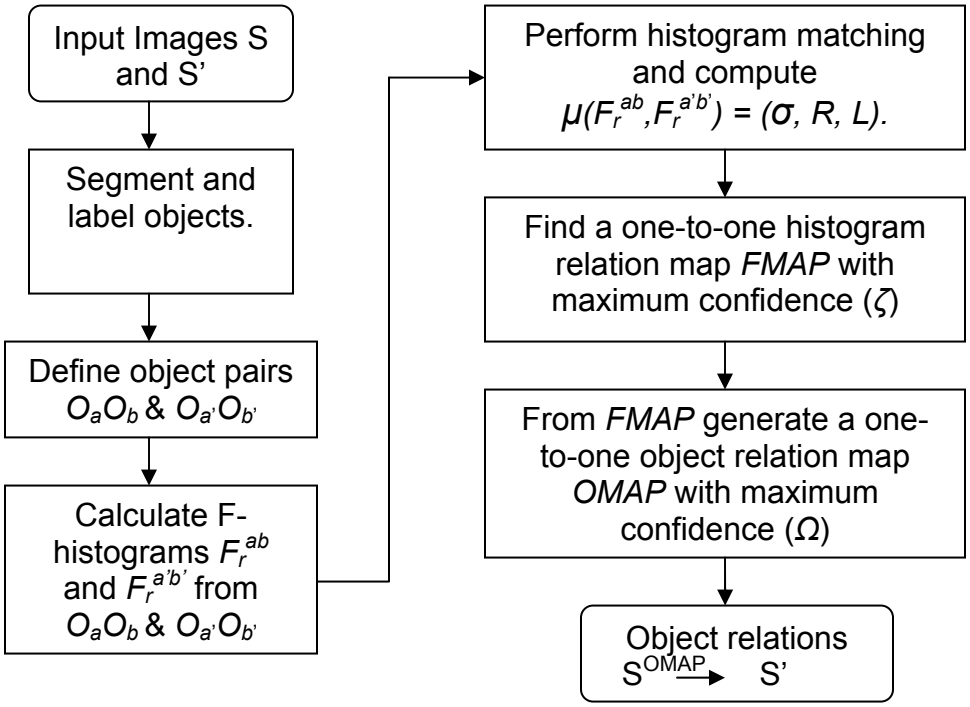


Figure 3.1. Overview of scene matching method [Sjahputera, 2004].

3.2 F-Histogram Calculation and Equalization

Once we have the input scenes the first step is to segment the two scenes. Segmentation on the sketchpad is handled by considering each object as a closed polygon [Bailey, 2003] [Skubic et. al., 2003]. Segmentation of the OGM is accomplished through a series of image processing operations that are discussed in chapter 4 under the section for experiment 4. The segmentation of

the sketch and the OGM is performed to extract the object boundary points, which are then used for computing the histogram of forces.

After segmentation, different object pairs are defined and the force histograms are generated for these pairs. The force histogram method allows us to capture the spatial relations between different object pairs in a scene, as described in chapter 2. The technique described in [Matsakis and Wendling, 1999], was used to generate these histograms. There are two types of F-histograms, i.e., the histogram of constant forces and the histogram of gravitational forces. In this work we are interested in the global view of the scene; hence, the histogram of constant forces has been used. These F-histograms are used as matching elements to perform scene matching.

F-histograms are generated for all unique pairs of objects in a scene. A collection of such histograms that can be used to effectively describe the spatial relations between all the object pairs in a scene is called as a 'Scene Descriptor' [Sjahputera, 2004]. Thus, in our case there are two scene descriptors, i.e., one for the template scene (sketch) and one for the argument scene (map or OGM). They are represented by F_rD and F_rD' respectively. The goal then, is to obtain a one-to-one histogram relation map between F_rD and F_rD'. Such a map is known as an FMAP [Sjahputera, 2004]. However, this task is not very straight forward. The two scenes may differ in terms of orientation, the shape and size of the objects, and the objects might be shifted due to translation. These differences are also captured by the F-histograms. Hence, the first task is to neutralize these differences.

The differences mentioned above may arise due to the fact that the template scene (sketch) has been drawn by a human and the argument scene (OGM or map) has been built by a robot and/or they could be due to differences in the viewing perspective. In [Matsakis et. al., 2004] certain F-histogram transformations were proposed that would neutralize any differences between the histograms of two scenes that could arise due to differences in the sensor pose parameters (viewing perspective). These transformations can be given as follows:

Scaling: Compute the histograms between different pairs of objects in each scene. Then compute their means m and m' using the following formulae:

$$m = \frac{1}{n}\sum_{\theta} F_r^{ab}(\theta) \qquad m' = \frac{1}{n}\sum_{\theta} F_r^{a'b'}(\theta)$$

where,

F_r^{ab} is a histogram relation between two objects a and b in template scene (S),

$F_r^{a'b'}$ is a histogram relation between two objects a' and b' in argument scene (S'),

θ represents a direction and $\theta \in (0^\circ, 180^\circ)$ and

$n = 180$.

Calculate the scaling factor (ℓ) using the formula given below:

$$\ell = \left(\frac{m'}{m}\right)^{\frac{1}{3-r}}$$

where, $r = 0$ for histogram of constant forces and

$r = 2$ for histogram of gravitational forces.

In this work histogram of constant forces is used.

Then apply the scaling transformation:

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

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 any scaling differences between the scenes.

Orientation: 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, and

ρ gives the direction in which the histogram must be shifted to account for orientation differences. It is computed as follows:

$$\rho = \begin{cases} c' - c & \text{if } (c' \geq c) \\ (c' - c) + 360 & \text{otherwise.} \end{cases}$$

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 the similarity between different pairs of histograms is computed. There are various similarity measures in the literature [Pappis and Karacapilidis, 1993] [Wang et. al., 1995] [Santini and Jain, 1999]. In this work, the normalized cross-correlation index is used that is computed as follows [Sjahputera, 2004]:

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

where,

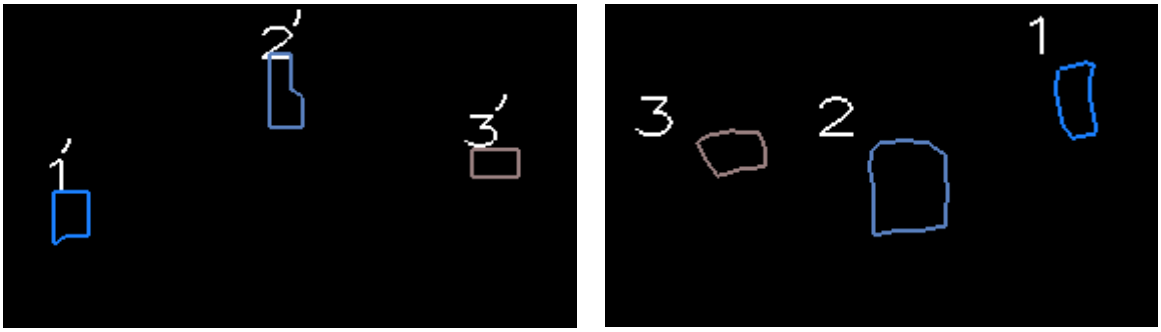
h_1 is $F_r^{a'b'}$ and h_2 is F_r^{a2b2}, and

$h_i(\theta)$ represents the histogram value for direction θ.

Finally, the 3-tuple output is saved after matching each pair of histograms:

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

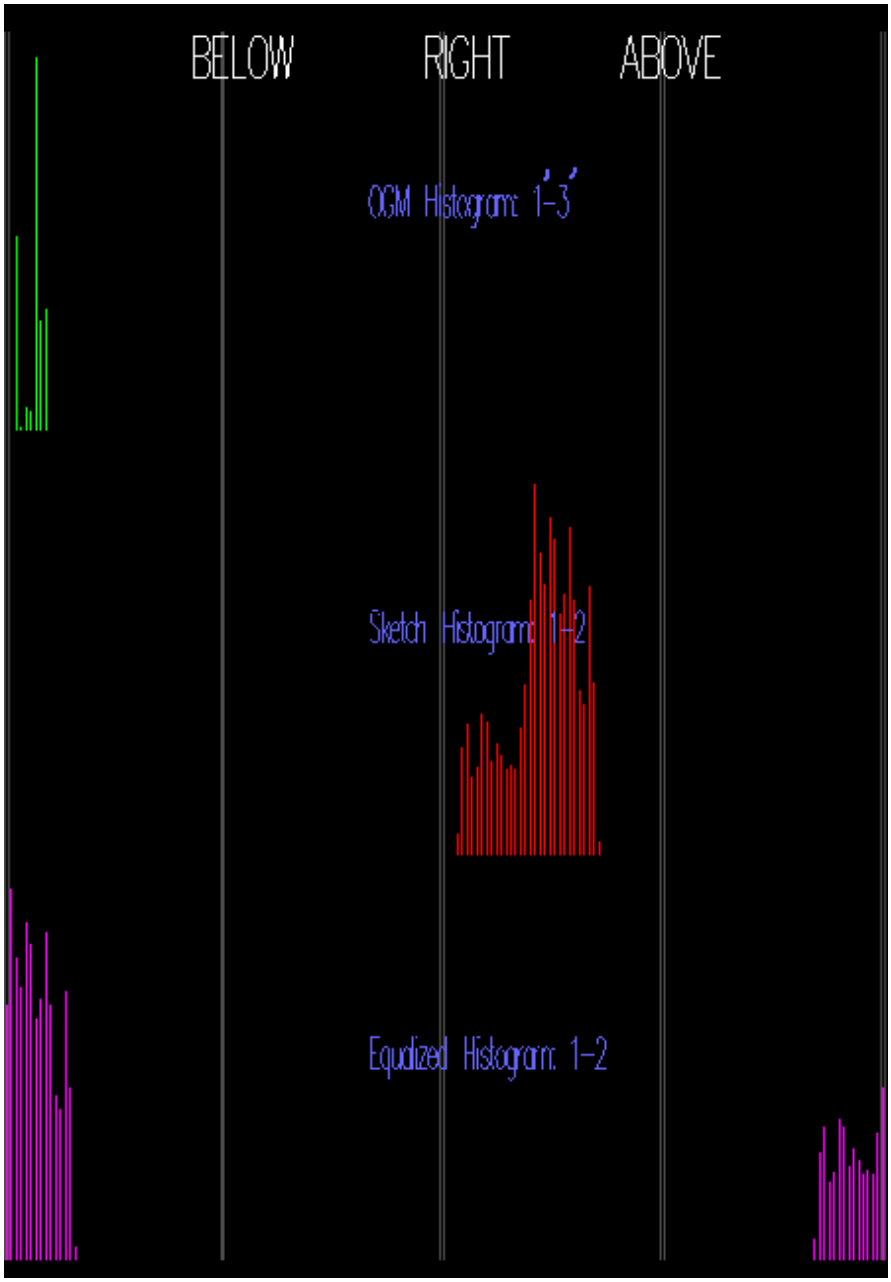
The above concepts can be explained with the help of the following example. Consider the OGM and the sketch shown below:



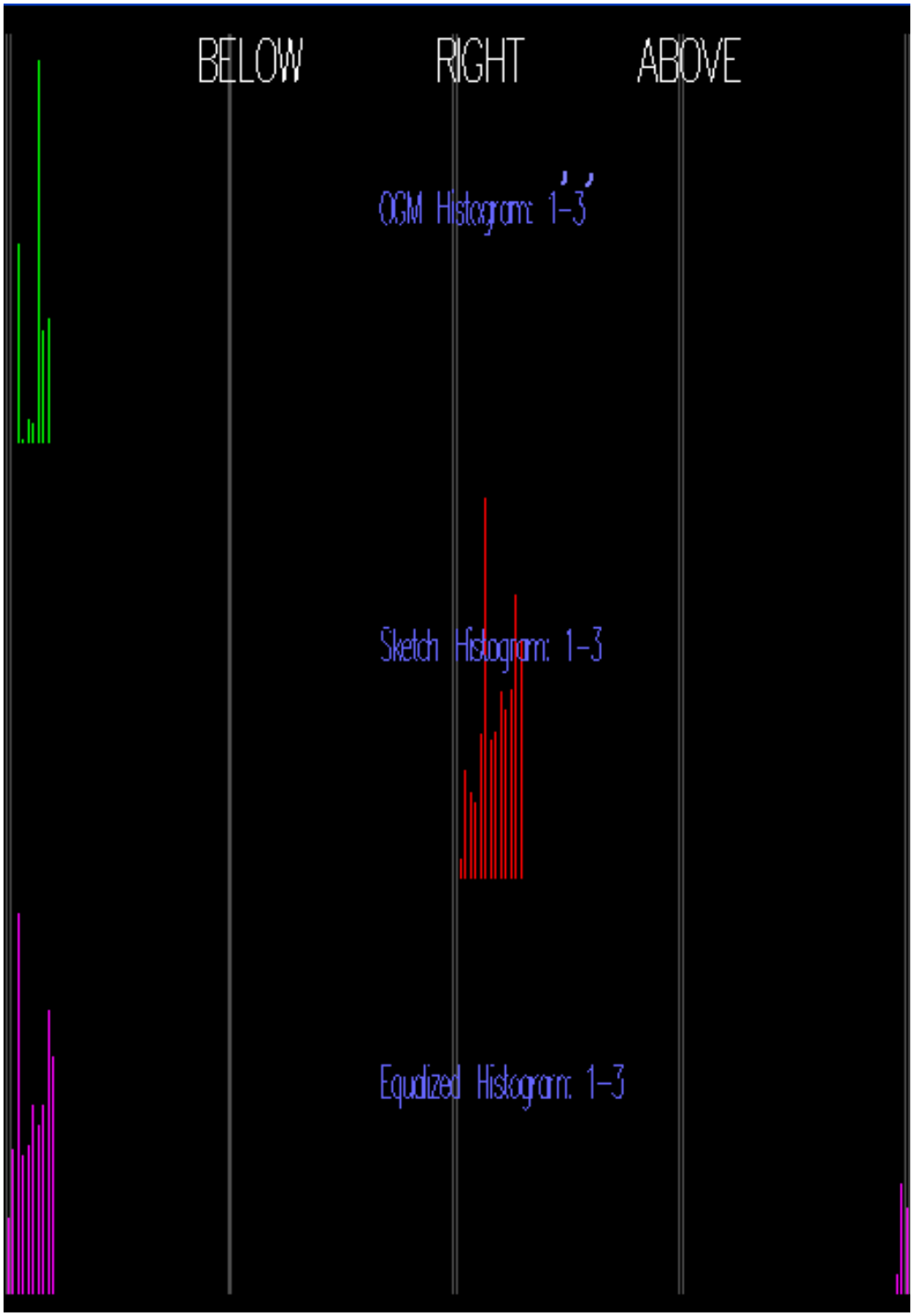
OGM Sketch

Figure 3.2. Example of scene matching between an OGM and a sketch.

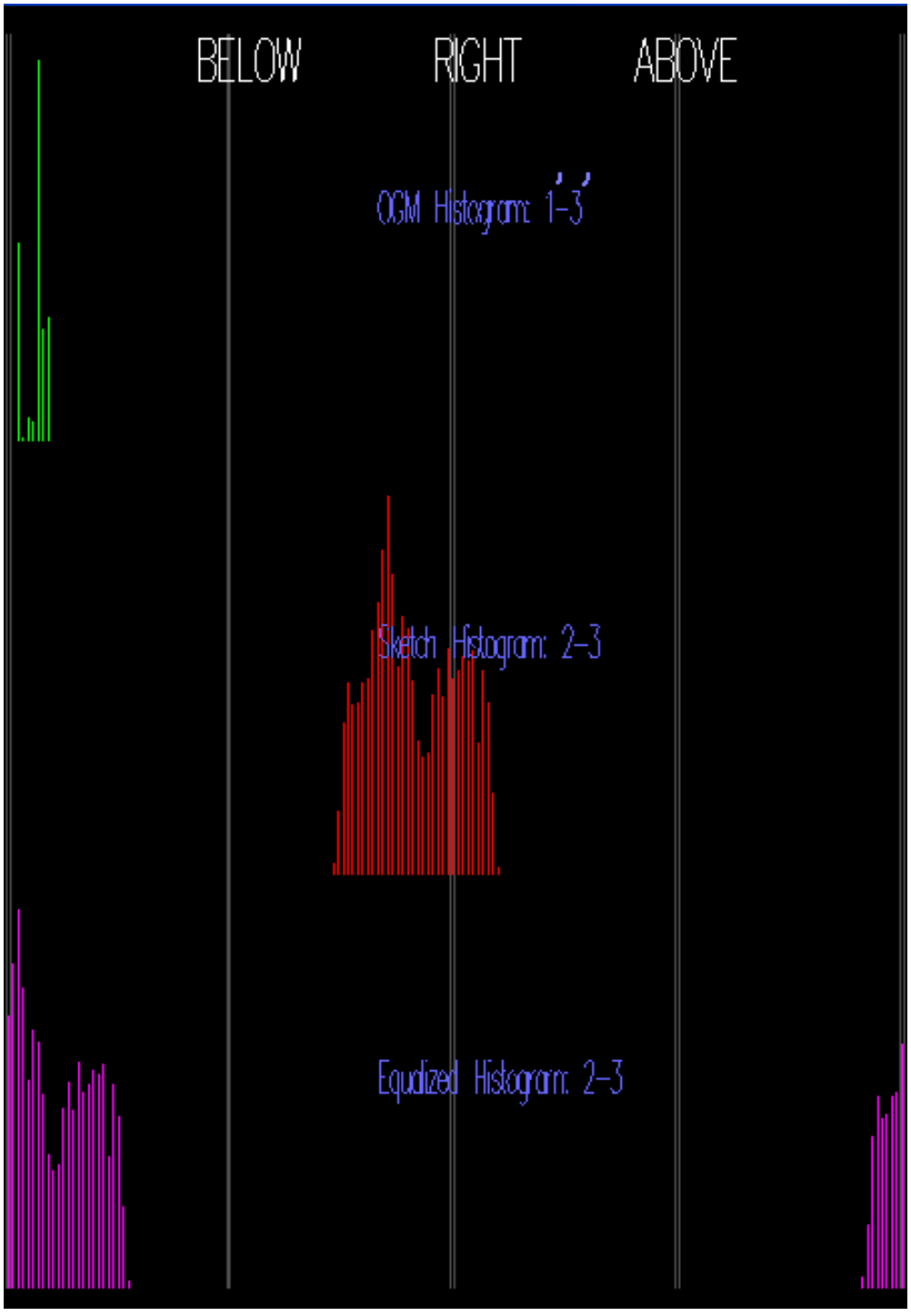
Objects 1', 2', and 3' in the OGM correspond to objects 1, 2, and 3 in the sketch respectively. The two representations have an orientation difference of 180°. The histogram of forces is generated for each object pair and then the process explained above is used to neutralize the scaling, orientation and translational differences. The results can be shown using the following figures:



(a)


BELOW
RIGHT
ABOVE
OGM Histogram: 1-3
Sketch Histogram: 1-3
Equalized Histogram: 1-3

(b)



(c)

Figure 3.3. Histograms Generated.

Figure 3.3. (a), (b), and (c) show three histograms each. Consider figure 3.3. (a), the top histogram is the histogram between objects 1' and 3' of the OGM. It shows that object 1' is to the left of and below object 3'. Similarly, the middle histogram is the histogram between objects 1 and 2 of the sketch. It shows that object 1 is to the right of and above object 2. This illustrates how the histogram of forces can be used to capture the spatial relations between different objects in a scene. Finally, the histogram at the bottom shows the middle histogram after transformation. In this histogram, scaling, orientation and translational differences are neutralized. Figure 3.3. (b) and (c) show the same with other histograms generated from the sketch.

3.3 Formation of 'Scene Descriptors'

The F-histogram method provides an effective way to capture the spatial relations information between a pair of objects in a scene. A 'scene descriptor' is made up of several such F-histograms that can define the spatial relations between all the objects in that scene [Sjahputera, 2004] [Sjahputera and Keller, 2005]. 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 the elements of F_rD and F_rD' are indexed based on the object labels:

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

where, $Q = N*(N\text{-}1) / 2$ and

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

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

where, $Q' = N'*(N'\text{-}1)$ and

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

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

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

The values for the indexes c, c', and c'' are always integers. This fact is used later, when the Evolutionary algorithm search is modeled to find the best FMAP.

Let $F_r^{(i)}$ represent an element in F_rD and $F_r^{(j')}$ represent an element in F_rD'. We consider all possible histogram mappings $F_r^{(i)} \rightarrow F_r^{(j')}$ defined by $F_rD \times F_rD'$. Each histogram mapping $F_r^{(i)} \rightarrow F_r^{(j')}$ is evaluated using the histogram matching algorithm that was previously described. This algorithm returns a 3-tuple output (σ, R, L). After normalization, this 3-tuple output gives us a measure of how good or bad the mapping $F_r^{(i)} \rightarrow F_r^{(j')}$ is. 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 [Sjahputera, 2004][Reddy and Chatterji, 1996]:

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.

3.4 FMAP Generator Algorithms

An FMAP is a one-to-one histogram correspondence map that matches a histogram in S with a histogram in S'. Finding a good FMAP is the most crucial step to build a one-to-one object map (OMAP) that has a high confidence (Ω). To be considered 'legal', an FMAP must satisfy the following requirements [Sjahputera, 2004]:

a. All the histogram relations in the FMAP must be one-to one.
b. Since F_rD' contains histogram relations and their duals, the second requirement is that an FMAP cannot contain a histogram relation AND its dual.
c. The final requirement is that the FMAP must lead to a consistent one-to-one object map (OMAP).

The above requirements make up the FMAP 'integrity' properties.

In this thesis three different approaches have been discussed to find a one-

to-one histogram relation map (FMAP) with the highest confidence (ζ). These are:

a. The Nearest Neighbor (NN) approach [Sjahputera, 2004] [Sjahputera and Keller, 2005].
b. The Fuzzy Sequential Nearest Neighbor (FSNN) approach [Sjahputera, 2004] [Sjahputera and Keller, 2005].
c. Evolutionary Algorithm for Scene Matching (EASM) approach.

The approaches (a) and (b) build the FMAP incrementally while searching for the best FMAP. Hence, they require two algorithms to work simultaneously. One algorithm builds the FMAP incrementally (NN or FSNN approach) and the other algorithm explores the search space and finds the point from where the optimal FMAP can be built (Particle Swarm Optimization).

In contrast, the Evolutionary Algorithm for Scene Matching (EASM) does not require two different algorithms to work simultaneously. An evolutionary algorithm is used to explore the search space. A population is evolved where each individual represents a 'legal' FMAP. Thus the FMAP is not built incrementally but is built as a whole.

Sections 3.5 and 3.6 explain the NN and FSNN approaches that are used to build the FMAP incrementally. Section 3.7 explains the particle swarm optimization algorithm that is used to find the optimal FMAP. Section 3.8 explains the EASM approach and finally sections 3.9 and 3.10 explain how a one-to-one object mapping OMAP is achieved from an FMAP.

3.5. The Nearest Neighbor (NN) Approach

In this approach, the FMAP is built using a clustering strategy [Sjahputera and Keller, 2005]. To start off, random seed points are selected and then the goal is to find the set of feature vectors that are closest to these points. Euclidean distance is used to measure the distance between a feature vector and the seed point (equation 3.5.1). Only one element is added to the FMAP at a time, i.e., the feature vector that is the closest to the seed point and satisfies the FMAP 'integrity' properties. The normalized 3-tuple output is used as the feature vector $\vec{x}_{ij'} = [\hat{\sigma} \quad \hat{R} \quad \hat{L}]$. 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 distance between two vectors where $\vec{x}_{ij'}, \vec{x}_{pq'} \in X$. Then this distance can be computed as follows:

$$d(\vec{x}_{ij'}, \vec{x}_{pq'}) = [(\hat{\sigma}_{ij'} - \hat{\sigma}_{pq'})^2 + (\hat{R}_{ij'} - \hat{R}_{pq'})^2 + (\hat{L}_{ij'} - \hat{L}_{pq'})^2]^{1/2} \qquad (3.5.1)$$

Let $\vec{y} \in [0, 1]^3$ be the seed point. Then the FMAP is grown as follows:

1. Let FMAP be an empty 1-D array with Q elements.
2. Calculate $d(\vec{y}, \vec{x}_{ij'})$ for all $\vec{x}_{ij'}$ in X.
3. Sort $d(\vec{y}, \vec{x}_{ij'})$ in ascending order. Call the sorted vector $\vec{x}_{ij'}^{(h)}$, for $h = 0, \ldots, (QQ' - 1)$. Initialize $h = 0$.
4. **Do While** FMAP is not full.

   4.1 **If** $\vec{x}_{ij'}^{(h)}$ satisfies the following condition:

   FMAP [i] *is empty* AND FMAP [k] $\neq j'$ AND

   {FMAP [k] $\neq j' - (Q'/2)$ (for $(Q'/2) \leq j' < Q'$) } OR

{FMAP [k] ≠ $j' + (Q'/2)$ (for $0 \le j' < (Q'/2)$) }

$k \in \{0, \ldots, Q-1\}$

Then

4.1.2 Add $\vec{x}_{ij'}^{(h)}$ to FMAP, i.e., FMAP[i] = j'.

End If (Step 4.1)

4.2 $h = h + 1$

End While (Step 4)

This method works well only if the correct histogram mappings are clustered in a tight hypersphere around the seed point. However, with less than ideal scenes (as is expected in hand-drawn sketches) this condition may not prevail. This was confirmed by the initial results where the NN method did not do well in this application.

3.6. The Fuzzy Sequential Nearest Neighbor (FSNN) Approach

This approach is a variation of the NN approach [Sjahputera and Keller, 2006]. Again, the histogram mappings are added one by one to gradually build a complete FMAP, but, instead of considering only the distance between the feature vector and the seed point we also consider the distance between the feature vector and other feature vectors that have already been added to the FMAP up to that point. This approach was found to be more robust than the NN approach.

In this method the fitness values of FMAP candidate vectors are calculated based on their "closeness" to vectors already assigned to the FMAP [Sjahputera,

2004][Sjahputera and Keller, 2006]. The degree of "closeness" is assessed using the fuzzy membership function that can be given by:

$$\mu_{close}(d) = \frac{1}{1 + e^{\alpha \times (d - m + k\upsilon)/\upsilon}} \tag{3.6.1}$$

where α controls the steepness of the membership function (we use α = 10). Also, we use k = 1. Parameters 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:

$$m = \frac{1}{QQ'(Q-1)(Q'-2)} \times \sum_{i=0}^{Q-1} \sum_{j'=0}^{Q'-1} \sum_{v=0, v \neq i}^{Q-1} \sum_{\substack{z'=0, z' \neq j', \\ z' \neq j' + (Q'/2), \\ z' \neq j' - (Q'/2)}}^{Q'-1} d(\vec{x}_{ij'}, \vec{x}_{vz'}) \tag{3.6.2}$$

$$\upsilon = \frac{1}{QQ'(Q-1)(Q'-2)} \times \sum_{i=0}^{Q-1} \sum_{j'=0}^{Q'-1} \sum_{v=0, v \neq i}^{Q-1} \sum_{\substack{z'=0, z' \neq j', \\ z' \neq j' + (Q'/2), \\ z' \neq j' - (Q'/2)}}^{Q'-1} [d(\vec{x}_{ij'}, \vec{x}_{vz'}) - m]^2 \tag{3.6.3}$$

The candidate vector with the best fitness that is in compliance with the FMAP integrity properties is added to FMAP. The FSNN performed better than the NN method on real world images by as much as 5% in [Sjahputera and Keller, 2006]. Based on this result, we use the FSNN method to generate our FMAP. The FSNN algorithm is given below:

Let $\vec{y} \in [0, 1]^3$ be the seed point. Then *FMAP* is grown as follows:

1. Let *FMAP* be an empty 1-D array with Q elements.
2. Calculate m and v using the equations (3.6.2) and (3.6.3).
3. Calculate $d(\vec{y}, \vec{x}_{ij'})$ for all $\vec{x}_{ij'} \in X$ using equation (3.5.1).

4. $\vec{x}_{IJ'} = min\ (\vec{x}_{ij'})$, add $F_r^{(I)} \rightarrow F_r^{(J')}$ to map, $FMAP[I] = J'$, $\vec{x}_{vz'}^{(0)} = \vec{x}_{IJ'}$.

5. $C(0) = \mu_{close}\ (d(\vec{x}_{vz'}^{(0)},\ \vec{y}\ ))$ as per equation (3.6.1).

6. $A = 1$ (stores the number of elements currently in *FMAP*).

7. **Do Until** *FMAP* is full (A=Q)

   7.1 **If** $\vec{x}_{ij'} \in X$ satisfies the following condition:

   FMAP [i] *is empty* AND *FMAP* [k] ≠ *j'* AND

   {*FMAP* [k] ≠ *j'* – *(Q'/2)* (for *(Q'/2)* ≤ *j'* < *Q')* }OR

   {*FMAP* [k] ≠ *j'* + *(Q'/2)* (for *0* ≤ *j'* < *(Q'/2)*) }

   $k \in \{0, \ldots, Q-1\}$

   THEN

   Compute the aggregate confidence measure ($CA_{ij'}$) using the following equation:

$$CA_{ij'} = \frac{1}{A} \times \sum_{a=0}^{A-1} C(a) \times \mu_{close}(d(\vec{x}_{ij'}, \vec{x}_{vz'}^{(a)}))$$

   where, $C(a)$ is the aggregate confidence measure obtained by the a^{th} vector added to the *FMAP*.

   7.2 Pick the best candidate vector $\vec{x}_{vz'}^{(A)}$ using the following equation:

$$\vec{x}_{vz'}^{(A)} = \max_{\vec{x}_{ij'}} (CA_{ij'})$$

$$C(A) = max\ (CA_{ij'})$$

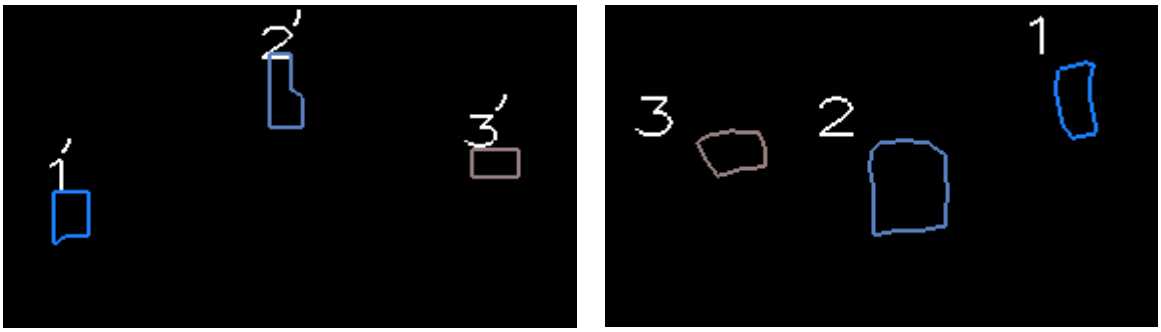
   7.3 Add $\vec{x}_{vz'}^{(A)}$ to map: $FMAP[v^{(A)}] = z'^{(A)}$

   7.4 $A = A + 1$

   End Until (Step 7)

The NN and the FSNN approach use particle swarm optimization to search for the optimal FMAP. Also in these approaches, only the 'integrity' properties 1 and 2 need to be satisfied for the FMAP to be considered 'legal'.

To further explain the above concepts, consider the example that was started in section 3.2. The following figure shows the OGM and the sketch.



OGM Sketch

Figure 3.4. Example of scene matching between an OGM and a sketch.

From the OGM and the sketch, the following histograms are generated.

Sketch (F_rD)	OGM (F_rD')	
1-2 (0)	1'-2' (0')	2'-1' (3')
1-3 (1)	1'-3' (1')	3'-1' (4')
2-3 (2)	2'-3' (2')	3'-2' (5')

The numbers in the parentheses represent the histogram indices in their respective scene descriptors. The goal is to find the correct correspondence between the scene descriptors, i.e. the correct FMAP, $F_rD(0) \longrightarrow F_rD'(0')$, $F_rD(1) \longrightarrow F_rD'(1')$, and $F_rD(2) \longrightarrow F_rD'(2')$. Each correspondence results in the 3 tuple output $(\hat{\sigma}, \hat{R}, \hat{L})$ which is used to asses the quality of the correspondence. There are several possible FMAP's, and the particle swarm optimization (PSO) or evolutionary algorithm is used to find the best FMAP that satisfies the FMAP

'integrity' requirements. The PSO algorithm provides several random seed points to the NN or FSNN algorithm and these algorithms build an FMAP incrementally from the seed points, as illustrated in Figure 3.5.

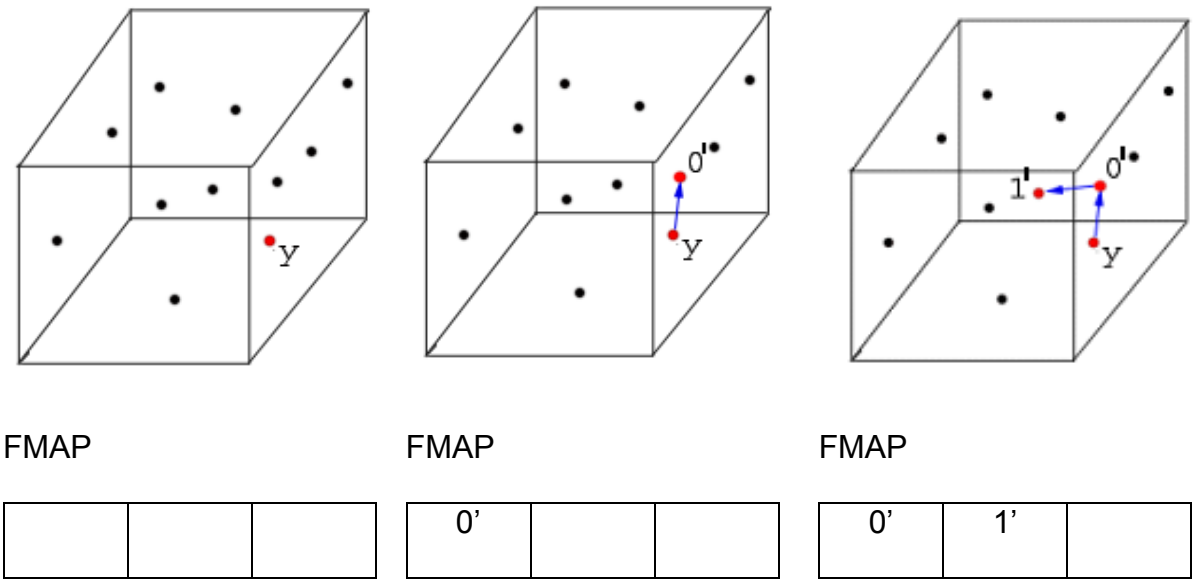


Figure 3.5. NN and FSNN approach

Both the NN and FSNN algorithms start with a random seed point y. The black dots represent the different feature vectors in 3-D space $(\hat{\sigma}, \hat{R}, \hat{L})$. The feature vector that is closest to the seed point and satisfies FMAP 'integrity' requirements is the first feature vector to be added to the FMAP. The Euclidean distance is used to compute the distance between the seed point and the feature vectors.

3.6.1 Evaluating the fitness of an FMAP (ζ):

Once an FMAP is generated (using any of the methods mentioned above) the next step is to evaluate its fitness. This is done by calculating the degree of confidence for an FMAP [Matsakis et al., 2004]. Three features are used for calculating the degree of confidence (ζ):

- Histogram Similarity based feature (ζ^{σ}): This feature measures the average histogram similarity across an FMAP. It is calculated as follows:

$$\zeta^{\sigma} = \frac{1}{Q} \times \sum_{k=0}^{Q-1} \sigma_k$$

  If all histogram comparisons in FMAP yield a perfect match then ζ^{σ} = 1.

- Rotation Angle based feature (ζ^{R}): This feature measures the consistency of estimated rotation angles R_k, i.e., if all R_k resulting from the histogram matching defined in FMAP yield the same values, then ζ^{R} = 1. We calculate ζ^{R} as follows:

$$R = \{\theta_k \mid k = 0,\ldots., Q;\ \theta_k \in [0, 360]\}$$

$$y_R = \frac{1}{Q} \sum_{k=0}^{Q-1} sin(\theta_k) \qquad x_R = \frac{1}{Q} \sum_{k=0}^{Q-1} cos(\theta_k)$$

$$\zeta^{R} = (x_R^2 + y_R^2)^{1/2}$$

- Dilation Ratio based feature (ζ^{L}): It can be computed as follows:

$$\zeta^{L} = e^{-K\,\delta L}$$

  where, $K = ln(2)$

$$\delta L = \frac{2}{Q \times (Q-1)} \times \sum_{k=0}^{Q-2} \sum_{\ell=k+1}^{Q-1} \left| \hat{L}_k - \hat{L}_\ell \right|$$

δL gives the average difference between any two normalized dilation ratios. If FMAP contains all correct histogram relations and all L are estimated perfectly, then $\delta L = 0$ and $\zeta^L = 1$; ζ^L decreases gradually as δL increases.

- The features ζ^σ, ζ^R, and ζ^L are fused to obtain $\zeta \in [0, 1]$ as the degree of confidence (or fitness) of a histogram map FMAP. The fusion result is calculated as the weighted average of the four features:

$$\zeta = \frac{w_1\zeta^\sigma + w_2\zeta^R + w_3\zeta^L}{w_1 + w_2 + w_3}$$

It is assumed that $w_1 = w_2 = w_3 = 1$. Thus, the above is a simple average.

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. [Matsakis et al., 2004], 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 images are less than ideal (as is expected in our case), then a correct FMAP is likely to have $\zeta < 1$, but ζ should be close to 1.

3.7 Search for the best FMAP

In order to obtain the correct one-to-one object mapping it is vital to first obtain the correct one-to-one histogram mapping. Of course, given the complexity of the search space, it is impossible to say with absolute certainty

whether or not a given histogram mapping is the correct one. Hence, a degree of confidence (ζ) is associated with each histogram mapping and several mappings from S to S' are explored. The method used to search for the best FMAP depends on the method that was used to generate the FMAP. If the Nearest Neighbor (NN) or the Fuzzy Sequential Nearest Neighbor (FSNN) methods were used as the FMAP generator algorithms (FMG), then particle swarm optimization is used as the search algorithm. In the Evolutionary Algorithm for Scene Matching (EASM) method an evolutionary algorithm is used to search for the best FMAP.

3.7.1 Particle Swarm Optimization (PSO)

The Nearest Neighbor (NN) and Fuzzy Sequential Nearest Neighbor (FSNN) are methods for finding the optimal FMAP given a seed point $\vec{y}$. In order to find an FMAP with the highest possible confidence (ζ), we need to use a search algorithm. The particle swarm optimization technique is used to accomplish this task [Sjahputera and Keller, 2005][Kennedy and Eberhart, 1995].

The particle swarm optimization algorithm is a population based search algorithm that is loosely modeled on the social behavior of animal groups, e.g., a flock of birds or a swarm of bees. A swarm consists of a set of individuals (referred to as particles) that are 'flown' through high dimensional search space. Each particle represents a possible solution to the optimization problem. The position of each particle is influenced by its own experience and the experience or knowledge, of its neighbors. Each particle tries to emulate the success of other

particles.

The PSO is driven by the social interaction between the particles. Individuals (particles) within the swarm learn from each other, and based on the knowledge obtained, move towards more 'promising' regions in the search space. The manner in which the particles interact is determined by the formation of neighborhoods. Different kinds of neighborhoods have been defined and studied [Kennedy, 1999][Kennedy and Eberhart, 1999]. In this work, a star neighborhood is used. In the star-type neighborhood, each particle can communicate with every other particle. Hence, the best particle influences the movement of the entire swarm. Each particle tries to imitate the overall best particle and improves its own performance in the process. The exchange of social information between particles is modeled with a velocity equation. The parameters that were used for the PSO are given below:

(1) Neighborhood: A star neighborhood was used. 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 [Sjahputera, 2004][Sjahputera and Keller, 2005][Sjahputera and Keller, 2005] and [Sjahputera and Keller, 2006] particles are initialized using a selective initialization scheme. This was done to provide the PSO with the likely starting points or seed points from where it could 'grow' the FMAP. Such a scheme has not been used in this work. In this work, $\hat{\sigma}, \hat{R}$ 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. The method for computing it has been explained in section 3.6.1.

(5) Velocity Restriction: The maximum velocity with which a particle can move in any direction is constrained. This ensures that the particles do not simply '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 30 iterations.

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

$$\vec{v}_k(t) = \vec{v}_k(t-1) + \rho_1(\vec{x}_{pbest} - \vec{x}_k(t)) + \rho_2(\vec{x}_{gbest} - \vec{x}_k(t)) \qquad (3.7.1)$$

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:

$$\vec{x}_k(t) = \vec{x}_k(t-1) + \vec{v}_k(t) \qquad (3.7.2)$$

The algorithm is given below:

1 Set t = 0.

2 Initialize the position of k particles $\vec{x}_k(t)$ 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 using NN or FSNN method and ($\vec{x}_k(t)$) as seed point.

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 If (4.1.3)

4.1.4 **If** $\zeta >$ *gbest* **Then**

4.1.3.1 $gbest = \zeta$, $\vec{x}_{k,gbest} = \vec{x}_k(t)$.

End If (4.1.4)

4.2 **For** each particle **Do**

4.2.1 Update $\vec{v}_k$ using (3.7.1).

4.2.2 Update $\vec{x}_k$ using (3.7.2).

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). The goal now, is 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 a one-to-one object correspondence map known as the OMAP is constructed. The method for constructing the OCCM is described in section 3.9 and the OMAP is described in section 3.10.

3.8 Evolutionary Algorithm for Scene Matching (EASM)

In this section a novel method is proposed for finding the optimal FMAP. This approach, known as the Evolutionary Algorithm for Scene Matching (EASM) is based on spatial relations and an evolutionary algorithm for accomplishing this task. The spatial relations between different objects are captured using the F-histogram method and then an evolutionary algorithm is used as a guided search to find the best histogram relational map (FMAP).

In an evolutionary algorithm we initialize a population. Each individual in the population represents a possible solution to the search problem. Thus the population contains several possible FMAPs. A FMAP can be thought of as a 1-D array with Q elements. Hence, each individual can be thought of as a 1-D array with Q integer elements. We then use an evolutionary algorithm to evolve the entire population and find the fittest individual (i.e. the best FMAP). The parameters used for the evolutionary algorithm are given below:

(1) Individual: We want each individual to represent a potential solution to the optimization problem. Hence each individual has *Q* elements and represents a possible FMAP. Each element in the FMAP represents a histogram relation in S and the value for that element represents a histogram relation in S'. Thus each element can have any value ranging from 0 to (*Q'*–1). Also, since all the histogram relations in F_rD and F_rD' are indexed using integer values, the elements can also have only integer values.

(2) Search space: Q - dimensional.

(3) Initial Population: To uniformly represent the entire search space, the entire population is initialized randomly. The only restriction applied is that each individual must satisfy the FMAP 'integrity' properties. To ensure that all the individuals 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) was used as the fitness function. The FMAP confidence (ζ) was calculated based on the similarity of the parameters $[\hat{\sigma}, \hat{R}, \hat{L}]$ across all histogram correspondences. 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 [Matsakis et al., 2004], 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 images are less than ideal (as is expected in our case), then a correct FMAP is likely to have $\zeta < 1$, but ζ should be close to 1. The method to compute ζ is explained in section 3.6.1.

(5) Cross-over: To generate the offspring, the parents were selected using random selection. The cross-over rate (P_c) was set at 0.7. A uniform cross-over operator was used and while computing the mask, the probability of cross-over for each element (P_x) was set equal to 0.5. This allowed for sufficient exchange of genetic material between the individuals.

(6) Mutation: The offspring generated using cross-over would usually not satisfy the FMAP 'integrity' properties. Hence to make the offspring 'legal', mutation was carried out on any element that was in conflict with the 'integrity' properties. After carrying out mutation, the offspring was again checked to ensure that the FMAP 'integrity' properties were met. Mutation was also carried out directly by selecting the parents using random selection and setting the mutation rate (P_m) to 0.3. This allowed for new genetic material to be introduced in the search.

(7) Next generation: The FMAP confidence value (ζ) was used as the fitness function. The parents and the offspring were put in the same pool and based on the value of their (ζ), the fittest individuals were selected to survive to the next generation.

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

- The maximum number of generations is reached.
- The best solution is constant for 400 iterations.

The algorithm can be given as follows:

1. Initialize the population and make sure they adhere to the FMAP integrity properties.
2. Let generation = 0.

While (No Convergence)

3. Select 2 parents using random selection.
4. Perform cross-over:

   4.1 Generate a random number $\xi_1 \in [0,1]$. **If** $\xi_1 < P_c$ **Then**

   4.1.1 Compute the mask. For each element of an individual Do

   4.1.1.1 Generate a random number $\xi_2 \in [0,1]$. If $\xi_2 < P_x$ then set the mask bit to 1 else set the mask bit to 0.

   4.1.1.2 If the mask bit is 1 then swap the genetic material at that position between the parents i.e. the element values.

   4.1.1.3 Check the offspring generated to make sure that they satisfy the FMAP integrity requirements. If they do not then randomly change the value of the element that is causing the conflict until the FMAP integrity properties are satisfied.

   Else (step 4.1)

   4.1.2 Return the parents without any change.

5. Select a parent.
6. Perform mutation:

   6.1 Generate a random number $\xi_3 \in [0,1]$. **If** $\xi_1 < P_m$ **Then**

   6.1.1 Randomly select elements in the parent and mutate them by

replacing the existing element value by some random number.

6.1.2 Check the offspring generated to make sure that it satisfies the FMAP integrity requirements. If it does not then randomly change the value of the element that is causing the conflict until the FMAP integrity properties are satisfied.

Else (step 6.1)

6.1.3 Return the parent without any change.

7. Evaluate the fitness (ζ) of the parents and the offsprings as explained in section 3.6.1.
8. Based on the fitness values, select the fittest to survive to the next generation.
9. generation = generation +1.

End While

The next two sections explain how the optimal FMAP obtained using the above methods can be translated into a one-to-one object mapping (OMAP) that allows the user to easily identify the mapping / registration of objects across the scenes.

3.9 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) [Sjahputera, 2004] [Sjahputera and Keller, 2005]. 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.

3.10 One-to-one Object Map (OMAP)

Upon obtaining the completed OCCM matrix, the next step is to determine the one-to-one object correspondence map (OMAP) such that the object

correspondence confidence value (Ω) is maximized. To do this, the OMAP is initialized as an empty 1-D array with N elements. Now, this array has to be filled with object relations that had the highest confidence in the OCCM. The most confident object relation is picked and added to the FMAP. For each subsequent object relation (say $O_g \rightarrow O_{h'}$), two conditions are checked:

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

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

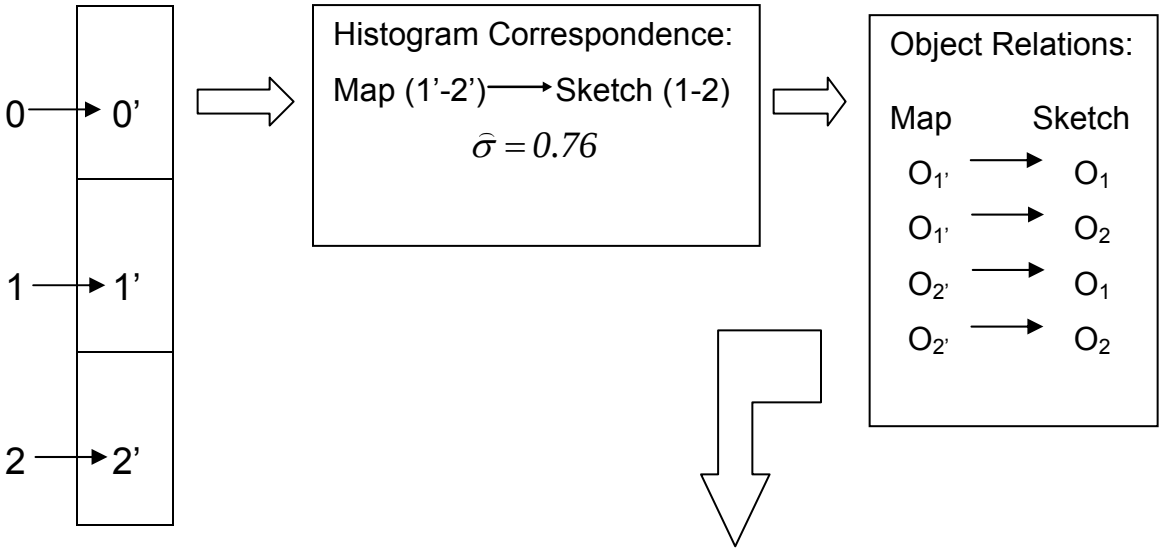
These conditions ensure that the one-to-one property of the OMAP is preserved. If the conditions are met then the object relation $O_g \rightarrow O_{h'}$ can be added to the OMAP and OMAP confidence (Ω) is incremented by $OC_{gh'}$. The OMAP thus obtained allows us to easily identify the correspondence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

The process of building an OMAP from an FMAP can be further explained with the help of our example. Suppose, at the end of the search process (PSO or EASM) we have the FMAP with the highest confidence; then the optimal OMAP can be built as illustrated in figures 3.6-3.8.

Figure 3.6 shows the first step in the process. Each element in the FMAP represents a histogram correspondence. Each histogram correspondence in turn represents four possible object relations. In this case, the OCCM is a 3×3 matrix. The histogram similarity measure ($\hat{\sigma}$) is distributed as partial evidence to support each of the four object relations. The process is repeated for each element in the FMAP as illustrated in figure 3.7 and 3.8. In the end, the object relations that

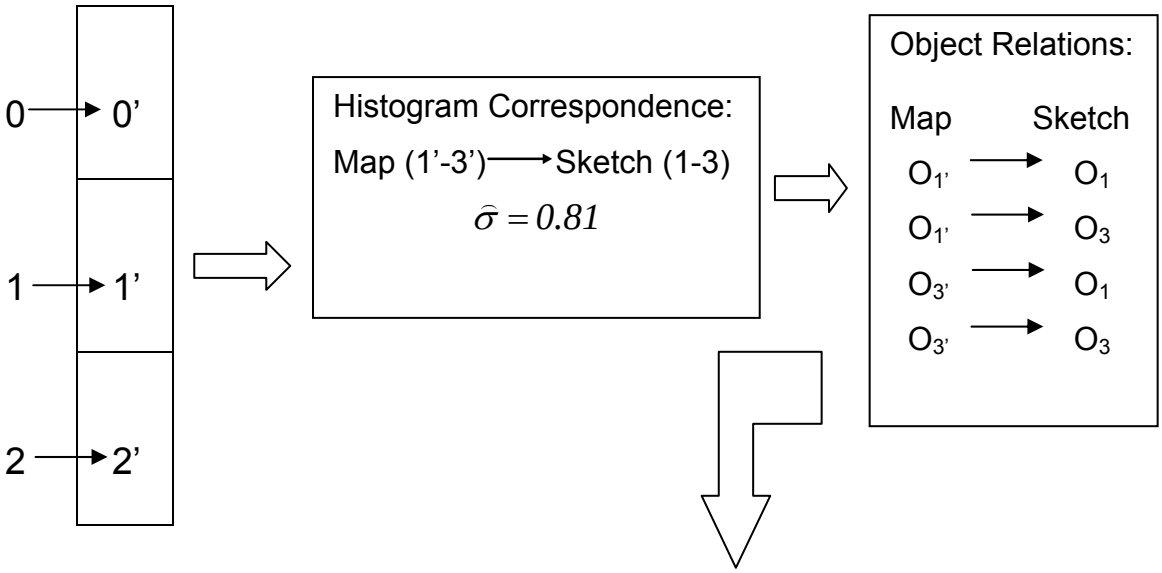
accumulate the highest amount of evidence and provide a one-to-one object mapping form the OMAP.

Optimal FMAP



Map \ Sketch	1	2	3
1'	0.76	0.76	
2'	0.76	0.76	
3'			

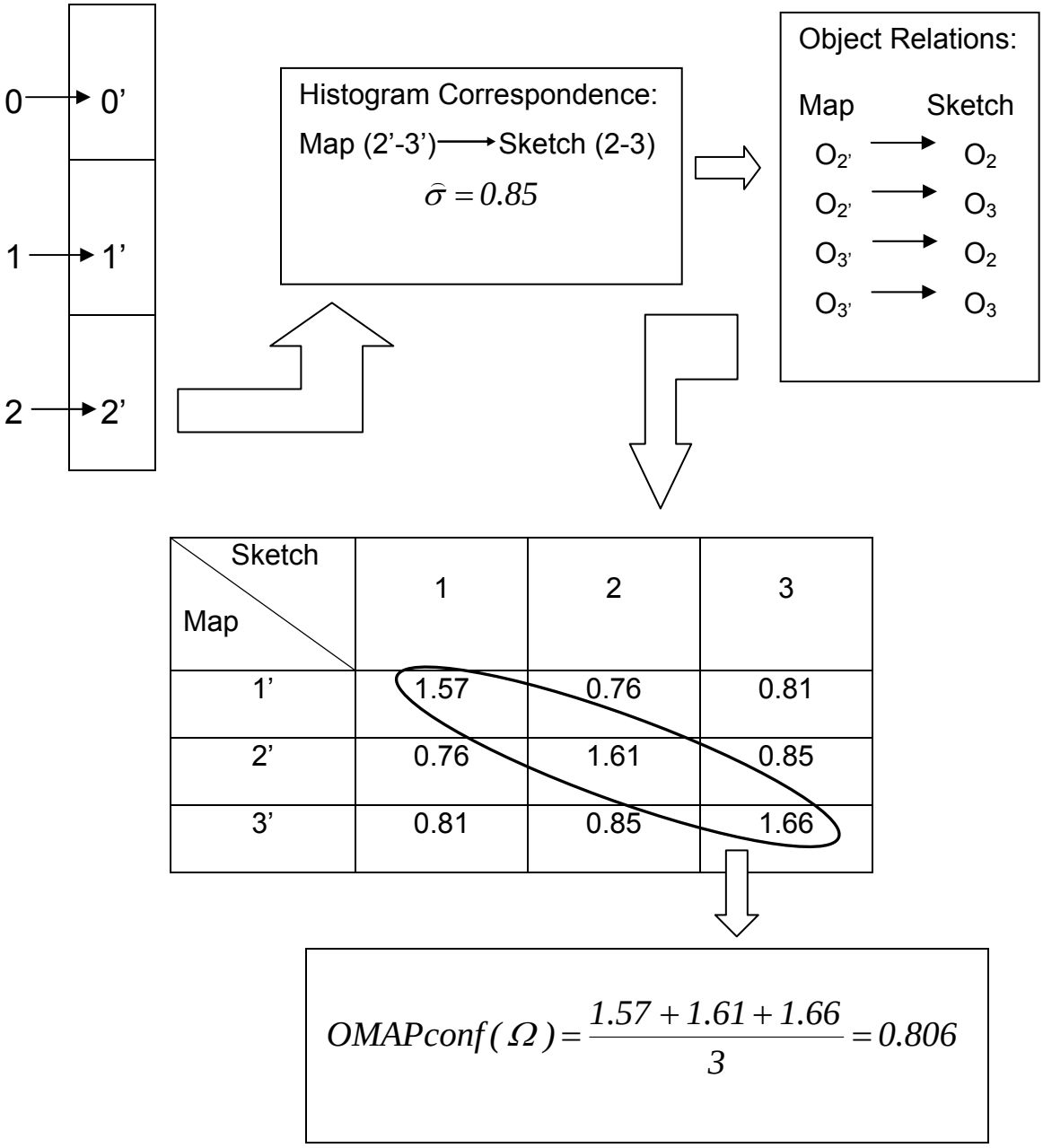
Figure 3.6. Step 1 for converting the optimal FMAP to an optimal OMAP.



Map \ Sketch	1	2	3
1'	1.57	0.76	0.81
2'	0.76	0.76	
3'	0.81		0.81

Figure 3.7. Step 2 for converting the optimal FMAP to an optimal OMAP.

Optimal FMAP



Map \ Sketch	1	2	3
1'	1.57	0.76	0.81
2'	0.76	1.61	0.85
3'	0.81	0.85	1.66

$$OMAPconf(\Omega) = \frac{1.57 + 1.61 + 1.66}{3} = 0.806$$

Figure 3.8. Step 3 for converting the optimal FMAP to an optimal OMAP and computing the OMAP confidence (Ω).

4 EXPERIMENTS AND RESULTS

To test the effectiveness and robustness of this approach, scene matching was performed between two types of quantitative maps and several hand drawn sketches. 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 [Bailey, 2003] [Skubic et al., 2003]. The scene matching algorithms were tested on the set of sketches and the physical map representing the scene. In addition, the approach was also tested by performing scene matching between a sketch and an occupancy grid map (OGM) of the same scene that was built by a robot using its sensors. The experiments have been categorized into five sets.

Experiment 1: Results using Fuzzy Sequential Nearest Neighbor (FSNN) on scenes A and B (physical map and the sketch have equal numbers of objects).

Experiment 2: Results using Evolutionary Algorithm for Scene Matching (EASM) on scenes A and B (physical map and the sketch have equal numbers of objects).

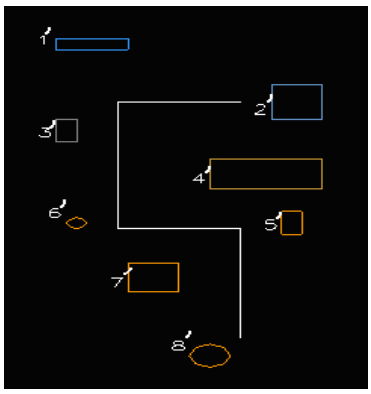
Experiment 3: Matching is performed between occupancy grid maps (OGM) built by a robot and the sketches for scenes A and B using EASM.

Experiment 4: Results using EASM on scenes A and B (physical map and the sketch have unequal numbers of objects).

Experiment 5: Matching is performed between OGMs and the sketches for scenes A and B using EASM (OGM and the sketch have unequal numbers of objects).

The scenes and the sketches used for the experiments are shown in figure 4.1 and 4.2. The sketched paths and labels shown in the figures were not used in the scene matching.

Quantitative map and sketches for scene A:



Physical map of the environment drawn to scale

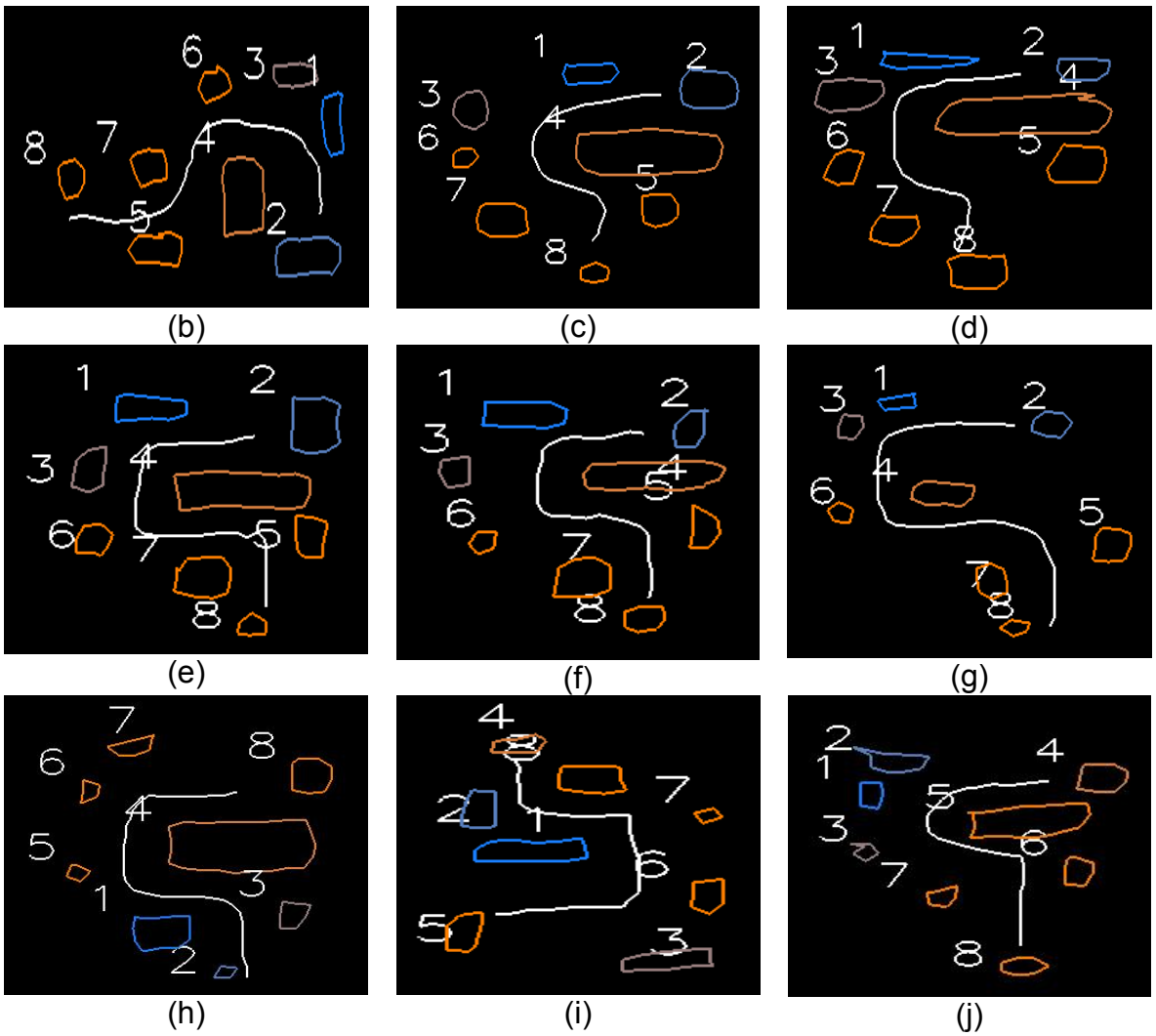
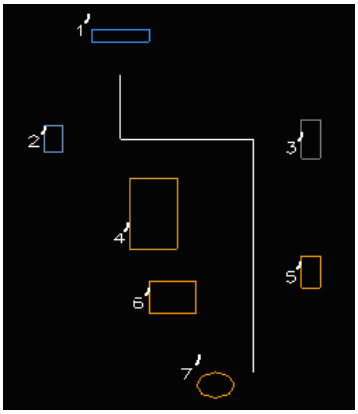


Figure 4.1. Quantitative map and the sketches for scene A

Quantitative map and sketches for scene B:



Physical map of the environment drawn to scale

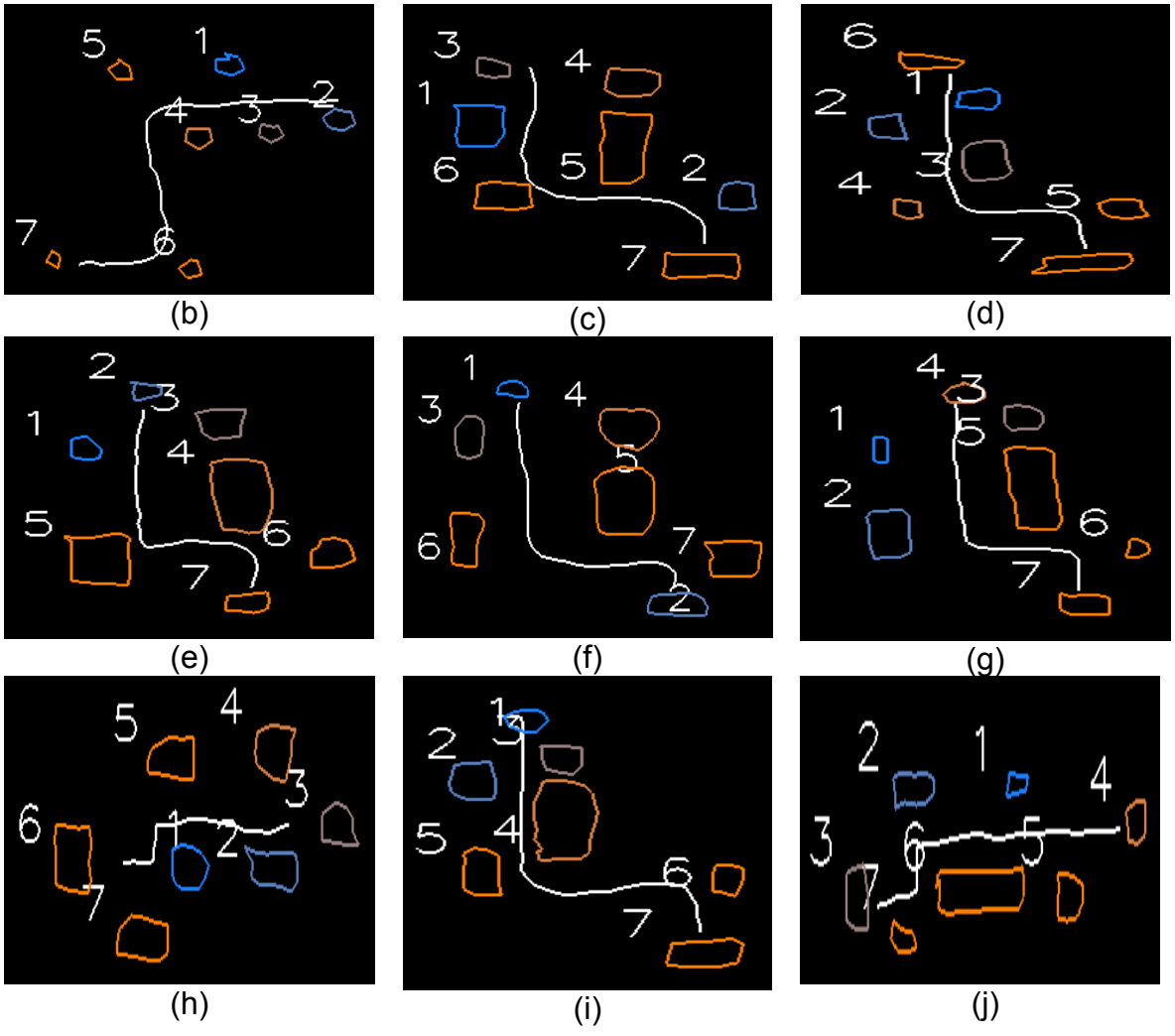


Figure 4.2. Quantitative map and the sketches for scene B

4.1 Experiment 1

In this experiment, scene matching was performed using the Fuzzy Sequential Nearest Neighbor (FSNN) approach in conjunction with the Particle Swarm Optimization (PSO) algorithm. Both the scenes A and B were matched with their respective sketches as shown in figures 4.1 and 4.2. The results have been reported in tables 4.1 and 4.2.

Table 4.1. Scene A FSNN Approach

Sketch	Confidence values for		Correct matches
	FMAP(ζ)	OMAP(Ω)	
a	1.000	1.000	8
b	0.881	0.528	8
c	0.874	0.570	8
d	0.860	0.519	6
e	0.909	0.718	8
f	0.907	0.752	8
g	0.922	0.694	8
h	0.904	0.752	8
i	0.879	0.659	8
j	0.916	0.737	8
Average	**0.905**	**0.693**	

Table 4.2. Scene B FSNN Approach

Sketch	Confidence values for		Correct matches
	FMAP(ζ)	OMAP(Ω)	
a	1.000	1.000	7
b	0.862	0.614	5
c	0.871	0.670	7
d	0.903	0.796	7
e	0.898	0.732	7
f	0.869	0.750	7
g	0.887	0.784	7
h	0.867	0.732	7
i	0.877	0.719	7
j	0.881	0.722	7
Average	**0.892**	**0.752**	

Analysis

From table 4.1 and 4.2, it can be concluded that the FSNN approach gave good results on all but one of the sketches for scenes A and B. Case (a) in the above tables represents an experiment where the quantitative map was matched with itself. This represents an ideal scenario where both the template and the argument scenes are the same. For this ideal scenario we got an ideal result with the FMAP (ζ) and OMAP (Ω) confidence equal to 1 and a perfect correspondence between all the objects.

Table 4.1 shows the results for scene matching between a map of scene A and the sketches that were collected from the user study and table 4.2 shows the same for scene B. The FSNN approach was successful in finding an FMAP with a high confidence value. This FMAP turned out to be the correct FMAP in most of the cases and resulted in the correct object mapping. Case (d) of scene A and case (b) of scene B represent cases in which the FSNN method could not get a perfect object mapping. To investigate these cases further, the objects were arranged such that the correct FMAP would be known. The fitness of the correct FMAP was then evaluated. In both the cases the fitness (ζ) of the FMAPs was found to be higher than the best fitness that the swarm could find. Thus, it was concluded that, if instead of the PSO, an exhaustive search had been used, 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

4.2 Experiment 2

In this experiment, scene matching was performed using the proposed Evolutionary Algorithm for Scene Matching approach (EASM). Both the scenes A and B were matched with their respective sketches as shown in figures 4.1 and 4.2. The results have been reported in tables 4.3 and 4.4. Table 4.5 shows the time taken for convergence for FSNN and EASM.

Table 4.3. Scene A EASM Equal Objects

Sketch	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
a	1.000	1.000	8
b	0.899	0.761	8
c	0.879	0.743	8
d	0.867	0.721	8
e	0.916	0.783	8
f	0.897	0.758	8
g	0.913	0.902	8
h	0.911	0.819	8
i	0.903	0.77	8
j	0.926	0.846	8
Average	**0.911**	**0.81**	

Table 4.4. Scene B EASM Equal Objects

Sketch	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
a	1.000	1.000	7
b	0.908	0.842	7
c	0.888	0.787	7
d	0.92	0.835	7
e	0.907	0.789	7
f	0.885	0.789	7
g	0.91	0.808	7
h	0.885	0.761	7
i	0.908	0.814	7
j	0.915	0.809	7
Average	**0.913**	**0.823**	

Table 4.5. Time for convergence

Objects in the Scene	**Time for convergence (in seconds)**	
	EASM Method	**FSNN Method**
3	1	**1**
4	4	**7**
5	6	**66**
6	12	**359**
7	22	**1424**
8	**37**	**4607**

Analysis

To compare Evolutionary Algorithm for Scene Matching (EASM) approach with Fuzzy Sequential Nearest Neighbors (FSNN) the same set of experiments were performed and the results were tabulated. Again, case (a) represents an experiment where the quantitative map was matched with itself. This is an ideal scenario and it gave an ideal result for both the scenes with the FMAP (ζ) and OMAP (Ω) confidence equal to 1 and a perfect correspondence between all the objects. Thereafter, scene matching was carried out between the quantitative map and the sketches that were collected from the user study [Bailey, 2003][Skubic et al., 2003]. The results can be seen in tables 4.3 and 4.4. A comparison of the convergence times is shown in table 4.5.

It is clear from tables 4.1 – 4.5 that the Evolutionary Algorithm for Scene Matching (EASM) approach offers several advantages over the Fuzzy Sequential nearest neighbor (FSNN) approach, namely:

(1) The algorithm can find FMAPs with a higher confidence.

(2) A more confident FMAP always translates into a more confident OMAP. The confidence value gives us a measure of the degree to which the algorithm is certain that it has found the correct FMAP or OMAP.

(3) From Table 4.5 it is clear that this algorithm offers a substantial speed up in

terms of the time required for convergence. This is considered to be one of the main advantages of this algorithm.

As can be seen from tables 4.1 – 4.4, in most of the cases, both the approaches found the same final object mapping; however, the EASM approach always associated a higher degree of confidence with its findings. Case (d) of scene A and case (b) of scene B represent cases where the FSNN method could not find the correct object mapping but the EASM method could find it.

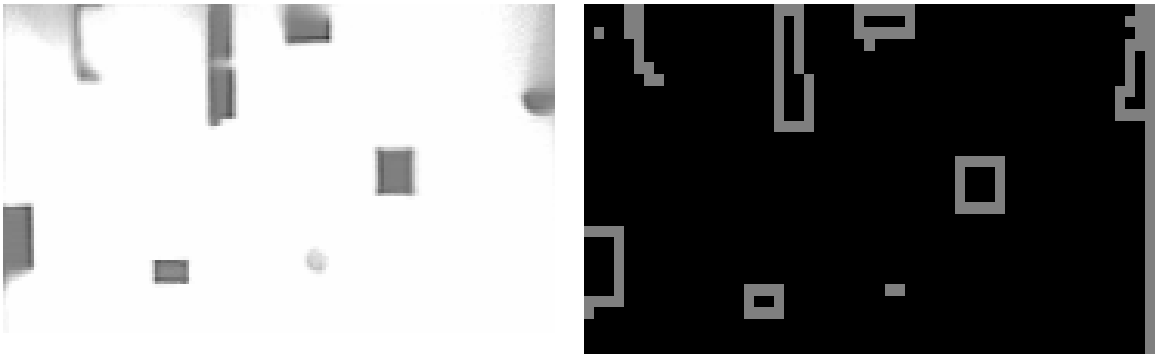
4.3 Experiment 3

The goal of this work was to develop an algorithm that would perform scene matching between an occupancy grid map (OGM) that had been built by a robot and a sketch of the environment that had been drawn by a novice user. This set of experiments was used to determine if this goal was met.

Before using the occupancy grid map (OGM) for scene matching, some image processing operations were performed on the sensory image map obtained from the robot. The image was first inverted and then a simple threshold was applied. Thereafter a boundary extraction operation was performed and the image was resampled so that the boundaries of all the objects would become straight lines. Figures 4.3 and 4.4 show the occupancy grid maps for both the scenes before and after they were processed. The chain code algorithm [Chakravarty, 1981] was used to extract the object boundary points from the scene, which were used to compute the force histograms that capture the spatial

relations between objects in a scene. Once the force histograms were available, scene matching was performed between the OGM and the various sketches.

Figure 4.3 and 4.4 show the OGMs for scenes A and B respectively. They also show the OGMs before and after processing. Tables 4.6 and 4.7 show the results obtained for scene A and B respectively.

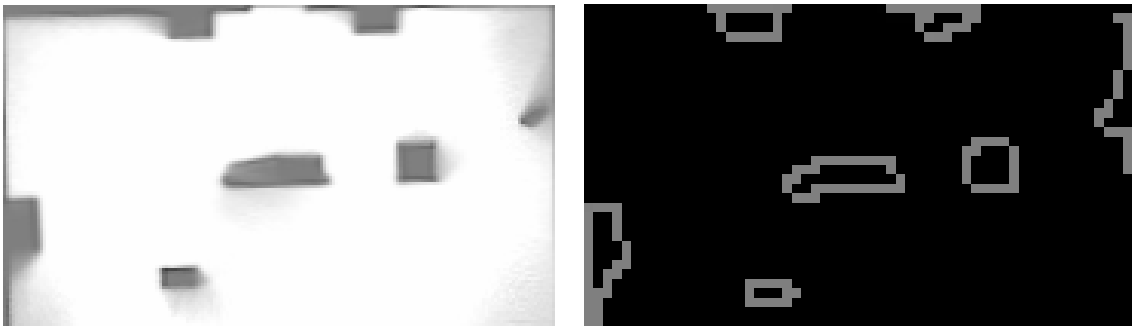


(a) (b)

Figure 4.3. (a) OGM obtained from the robot for scene A. (b) OGM after processing.

Table 4.6. Scene A EASM with OGM

Sketch	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
a	1.000	1.000	8
b	0.879	0.714	8
c	0.879	0.708	8
d	0.865	0.692	8
e	0.895	0.728	8
f	0.888	0.73	8
g	0.872	0.777	8
h	0.886	0.734	8
i	0.899	0.752	8
j	0.896	0.757	8
Average	**0.896**	**0.759**	



(a) (b)

Figure 4.4. (a) OGM obtained from the robot for scene B. (b) OGM after processing.

Table 4.7. Scene B EASM with OGM

Sketch	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
a	1.000	1.000	7
b	0.871	0.71	7
c	0.906	0.817	7
d	0.932	0.874	7
e	0.914	0.815	7
f	0.918	0.86	7
g	0.928	0.853	7
h	0.929	0.837	7
i	0.925	0.855	7
j	0.909	0.834	7
Average	**0.923**	**0.845**	

For scene A, when the sketches were matched with the physical map, the average FMAP and OMAP confidences were 0.911 and 0.81 respectively. When the sketches were matched with the OGM, the average FMAP and OMAP confidences were 0.896 and 0.759 respectively. For scene B, when the sketches were matched with the physical map, the average FMAP and OMAP confidences were 0.913 and 0.823 respectively. When the sketches were matched with the OGM, the average FMAP and OMAP confidences were 0.923 and 0.845 respectively. From the tables and the above discussion it is clear that the results

obtained from using the OGM are not very different from the ones obtained using the physical map. This is something that was assumed in the initial part of this work. The results obtained in tables 4.6 and 4.7 serve to validate that assumption.

4.4 Experiment 4

The aim of the sketch interface is to allow novice users to communicate with the robot and to enable them to perform some simple tasks using the robot. In such applications where the user has no experience with robotics, it is possible that he may forget to draw certain objects that are present in the scene or he may draw an extra object. The EASM algorithm accommodates such cases and returns the best one-to-one object mapping that it can find. This set of experiments is used to demonstrate this capability of the algorithm.

In order for the algorithm to execute correctly, it is required that the template scene (S) has either equal or fewer objects than the argument scene (S'). In all the previous cases the number of objects in the sketch and the map were always equal. However, since that is not the case in this set of experiments, care must be taken while selecting the template and argument scenes. The following are some sketches that were also collected in the same user study [Bailey, 2003] [Skubic et al., 2003]. It can be clearly seen that all of them contain fewer objects than the actual scene. Figure 4.5 shows the sketches for scene A and table 4.8 gives the results obtained for these sketches. Figure 4.6 shows the sketches for scene B and table 4.9 gives the results obtained for these sketches.

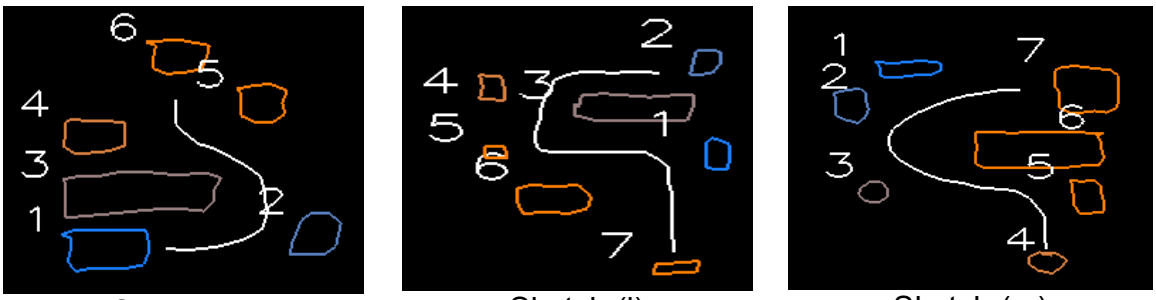


Figure 4.5. 3 more sketches of scene A with unequal number of objects.

Table 4.8. Scene A EASM unequal number of objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.899	0.769	5	Object 6 (Sketch) -> Object 7' (Map)
l	0.912	0.81	7	
m	0.885	0.739	7	

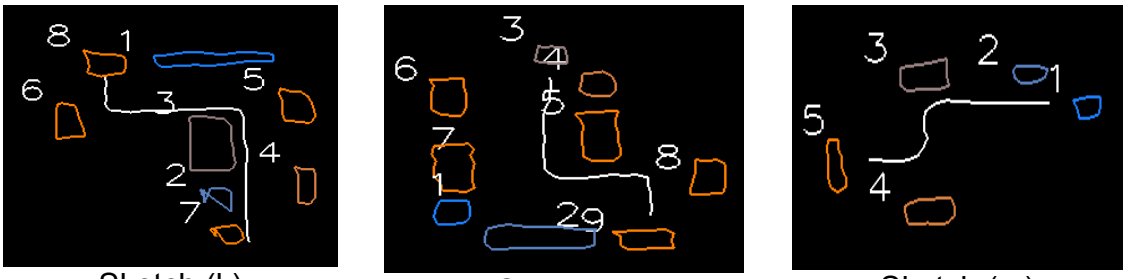


Figure 4.6. 3 more sketches of scene B with unequal number of objects.

Table 4.9. Scene B EASM unequal number of objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.899	0.755	7	
l	0.901	0.782	6	Object 7 (Sketch) -> Object 3' (Map)
m	0.902	0.788	4	Object 1 (Sketch) -> Object 6' (Map)

Sketch (k) of scene A has two missing objects. When this sketch is compared with the quantitative map, we find objects 1’ and 7’ missing in the sketch. The algorithm matches object 6 in the sketch with object 7’ in the map. Actually, object 6 in the sketch corresponds to object 8’ on the map.

If the spatial relations between the different objects are considered, then it is not a surprise that the algorithm makes a correspondence between object 6 on the sketch and object 7’ on the map. The algorithm generally returns the correct one-to-one mapping between the objects of the two scenes. However, there are certain cases in which it fails and those cases can be explained in a similar manner.

Another example is sketch (l) of scene B. In this case the sketch has more objects than the quantitative map. When scene matching is performed using this sketch, the algorithm makes a correspondence between object 7 of the sketch and object 3’ of the map. Actually, object 1 in the sketch corresponds to object 3’ in the map. When object 7 of the sketch is compared with object 3’ of the map it can be seen that both these objects share similar spatial relationships with all the other objects in the scene. Hence it is only logical that the algorithm makes a correspondence between these two objects.

To get a better understanding of how well the algorithm performs with missing objects, more test cases were required. Therefore, the sketches in which the user had drawn all the objects were also used. In these experiments, an object was removed from each sketch. The algorithm was run and the results were tabulated. The object was then inserted back and the next object was removed.

The test was repeated until each of the objects had been removed. Tables 4.10 and 4.11 show the results for these tests.

Table 4.10. Scene A sketch (g) – 1 Missing Object

Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.915	0.9	7	
2	0.908	0.898	6	Object 1 (Sketch) -> Object 2' (Map)
3	0.913	0.901	7	
4	0.927	0.903	7	
5	0.917	0.912	7	
6	0.903	0.892	7	
7	0.916	0.882	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.906	0.906	7	
Average	**0.913**	**0.899**		

Table 4.11. Scene B sketch (d) - 1 Missing Object

Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.913	0.856	5	Object 6 (Sketch) -> Object 6' (Map)
2	0.924	0.843	6	
3	0.919	0.81	6	
4	0.913	0.812	6	
5	0.918	0.83	6	
6	0.925	0.856	6	
7	0.94	0.887	6	
Average	0.906	0.817		

The results shown in tables 4.10 and 4.11 are representative examples of the typical results that were obtained for both the scenes. Detailed results for scene

A can be found in Appendix A and results for scene B can be found in Appendix B. Table 4.18 is a table that summarizes all the results and is discussed at the end of this chapter. From these results, the following conclusions can be drawn:

- The quality of the sketch plays an important role in determining how well the algorithm performs.
- The location of the object that is missing or is extra also plays an important role in determining whether or not we can get the correct mapping.

Table 4.12 shows the results for performing scene matching between sketch (e) of scene A and its quantitative map and table 4.13 shows the results for performing scene matching between sketch (e) of scene B and its quantitative map. In these experiments, the number of objects missing was incremented until the algorithm completely failed. It was used to test the breaking point of this algorithm. Sketches (e) of scene A and scene B were selected because the algorithm had performed very well for these sketches with one missing object. The tables show that even for a good sketch, the algorithm fails if the number of objects missing becomes equal to or greater than 5 for scene A and equal to or greater than 4 for scene B.

Table 4.12. Scene A sketch (e) – Several Missing

Number of Missing Objects	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.912	0.792	7
2	0.902	0.77	6
3	0.915	0.788	5
4	0.925	0.832	4
5	0.94	0.89	0

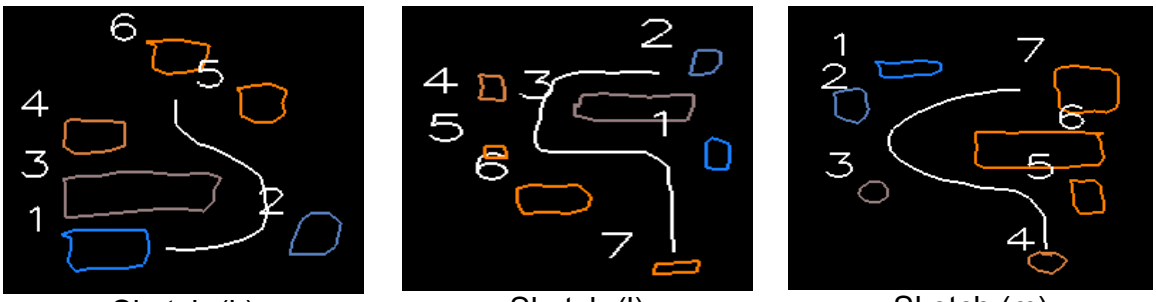
Table 4.13. Scene B sketch (e) - Several Missing

Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.913	0.774	6
2	0.918	0.791	5
3	0.923	0.803	4
4	0.935	0.852	0

4.5 Experiment 5:

The EASM algorithm accommodates for cases where the user may have forgotten to draw certain objects in the sketch that are present in the scene. In such cases the algorithm returns the best one-to-one object mapping that it can find. This set of experiments is used to demonstrate this capability of the algorithm by performing scene matching between an OGM and the sketches of scenes A and B, where the number of objects in the OGM is not equal to the number of objects in the sketch.

Figure 4.7 shows the sketches for scene A and table 4.14 gives the results obtained for these sketches. Figure 4.8 shows the sketches for scene B and table 4.15 gives the results obtained for these sketches.

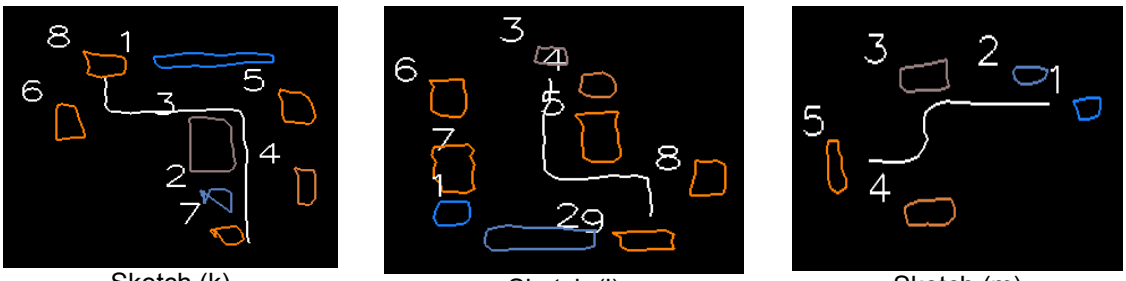


Sketch (k) Sketch (l) Sketch (m)

Figure 4.7. 3 more sketches of scene A with unequal number of objects.

Table 4.14. Scene A EASM unequal number of objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.88	0.731	6	
l	0.912	0.81	7	
m	0.866	0.745	6	Object 3 (Sketch) -> Object 7' (Map)



Sketch (k) Sketch (l) Sketch (m)

Figure 4.8. 3 more sketches of scene B with unequal number of objects.

Table 4.15. Scene B EASM unequal number of objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.916	0.826	7	
l	0.922	0.844	6	Object 7 (Sketch) -> Object 3' (Map)
m	0.902	0.788	3	Object 2, 3 (Sketch) ->Object 6', 4' (Map)

Sketch (m) of scene A has one missing object. When this sketch is compared with the quantitative map, we find object 7' is missing in the sketch. The algorithm matches object 3 in the sketch with object 7' in the map. Actually, object 3 in the sketch corresponds to object 6' on the map.

If the spatial relations between the different objects are considered, then it is not a surprise that the algorithm makes a correspondence between object 3 on the sketch and object 7' on the map. The algorithm generally returns the correct one-to-one mapping between the objects of the two scenes. However, this represents one of the cases in which it fails.

Another example is sketch (l) of scene B. In this case the sketch has more objects than the quantitative map. When scene matching is performed using this sketch, the algorithm makes a correspondence between object 7 of the sketch and object 2' of the map. Actually, object 1 in the sketch corresponds to object 2' in the map. When object 7 of the sketch is compared with object 2' of the map it can be seen that both these objects share similar spatial relationships with all the other objects in the scene. Hence it is only logical that the algorithm makes a correspondence between these two objects.

To get a better understanding of how well the algorithm performs with missing objects, more test cases were required. Therefore, the sketches in which the user had drawn all the objects were also used. In these experiments, an object was removed from each sketch. The algorithm was run and the results were tabulated. The object was then inserted back and the next object was removed. The test was repeated until each of the objects had been removed. Tables 4.16 and 4.17 show the results for these tests.

Table 4.16. Scene A sketch (g) – 1 Missing Object

Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.885	0.802	7	
2	0.88	0.792	7	
3	0.882	0.797	7	
4	0.897	0.801	7	
5	0.871	0.776	7	
6	0.865	0.754	7	
7	0.872	0.731	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.881	0.761	7	
Average	**0.879**	**0.776**		

Table 4.17. Scene B sketch (d) - 1 Missing Object

Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.922	0.876	5	Object 6 (Sketch) -> Object 5' (Map)
2	0.932	0.868	6	
3	0.94	0.89	6	
4	0.942	0.893	6	
5	0.932	0.888	6	
6	0.918	0.856	6	
7	0.935	0.898	6	
Average	**0.932**	**0.881**		

The results shown in tables 4.16 and 4.17 are representative examples of the typical results that were obtained for both the scenes. Detailed results for scene A can be found in Appendix C and results for scene B can be found in Appendix D. From these results, the following conclusions can be drawn:

- The quality of the sketch plays an important role in determining how well the algorithm performs.

- The location of the object that is missing or is extra also plays an important role in determining how the algorithm performs.

4.6 Comprehensive Summary for Experiments 4 and 5:

The comprehensive results for experiments with missing objects are given in table 4.18:

Table 4.18. Summary of results for experiments with missing objects

Scene	No. of Tests	No. of objects being matched	No. of tests with perfect results	No. of objects correctly matched	Average FMAP confidence (ζ)	Average OMAP confidence (Ω)
Scene A sketch with Map	72	504	62	493	0.897	0.799
Scene A sketch with OGM	72	504	65	497	0.905	0.812
Scene B sketch with Map	63	378	56	371	0.876	0.732
Scene B sketch with OGM	63	378	56	370	0.909	0.808
Total	270	1764	239	1731	0.897	0.788

Figure 4.1 shows 9 sketches of scene A that have eight objects each. For each test, an object is removed from a sketch, scene matching is performed and then the object is replaced. Thus 8 tests can be performed on each sketch, one per object. Since there are nine such sketches we have 9*8 = 72 such tests. In each test there are seven objects being matched. Therefore, the total number of objects being matched is 9*8*7 = 504. Similarly, for scene B we have 9*7 = 63 such tests and the total number of objects being matched is 9*7*6 = 378.

From the table it can be seen that the results obtained from using the OGM are not very different from the ones obtained using the quantitative map. The EASM approach finds the correct FMAP and OMAP with high confidence. Over all, the EASM approach gives good results for sketches with missing objects. The detailed results for each sketch can be found in appendices A-D.

5 CONCLUSION

In this work, a novel method was proposed to perform scene matching between a quantitative representation of the scene and a qualitative representation of the scene. A precise physical map or an occupancy grid map was used as the quantitative representation of the scene, and a hand drawn sketch map was used as the qualitative representation of the scene. Consideration was given to differences in scene orientation, shape and sizes of the objects and the translation of the objects. The force histogram method was used to capture the spatial relations between different object pairs in a scene.

Several approaches were discussed for obtaining the correct one-to-one histogram correspondence map (FMAP) and the correct one-to-one object correspondence map (OMAP). These included using the Nearest Neighbor (NN) or the Fuzzy Sequential Nearest Neighbor (FSNN) approach [Sjahputera, 2004] in conjunction with the particle swarm optimization algorithm (PSO) [Sjahputera and Keller, 2005], or using the proposed Evolutionary Algorithm for Scene Matching (EASM). Scene matching was performed for two different scenes between a physical map and a sketch and the results were reported. The EASM approach was found to give superior results in terms of the correct object mappings, the FMAP and OMAP confidences, and in terms of the time it took for convergence. Scene matching was performed between an occupancy grid map and a sketch for both the scenes and the results were reported. When the template and the argument scene had equal number of objects the algorithm

found the correct mapping with 100% accuracy. To test the approach further, scene matching was also performed on cases where the quantitative representation of the scene did not have the same number of objects as the qualitative representation. Such cases typically arise when the user forgets to draw an object in the sketch or draws an extra object in the sketch. Comprehensive tests were done for the case of one missing object and the results were reported in chapter 4 and Appendices A-D. For scene A, the algorithm was accurate 88.2% of the times and for scene B the accuracy was 88.8%.

From the results it is clear that the EASM approach proposed in this work offers a tremendous advantage over the other approaches. It is fundamentally different from the other approaches in the way it searches for the best FMAP. Also, from the results it is clear that in the case of unequal numbers of objects, the quality of the sketch plays an important role in determining if the algorithm finds the correct one-to-one object mapping. Over all, the results are very encouraging.

In the future, efforts will be directed towards integrating this scene matching algorithm with the sketch interface application. This would enable the robot to perform scene matching in real time. Also, an alternative approach is being investigated in which scene matching would be performed by matching the objects directly.

APPENDIX A

Scene matching between scene A (physical map) and sketches with unequal number of objects

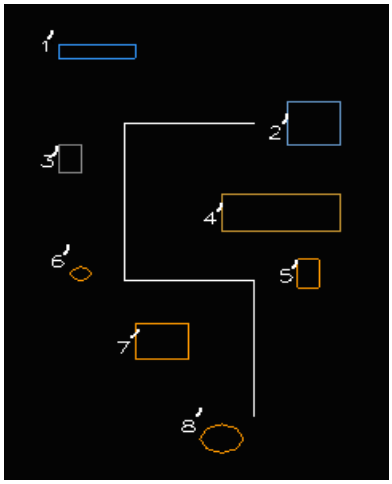
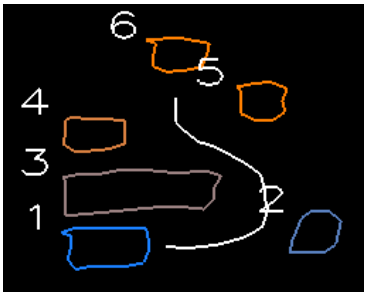
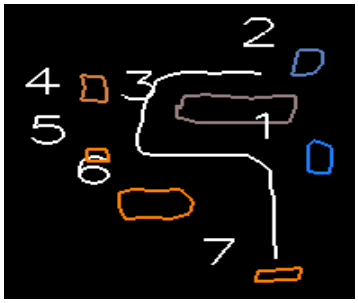


Figure A.1. Physical map of scene A

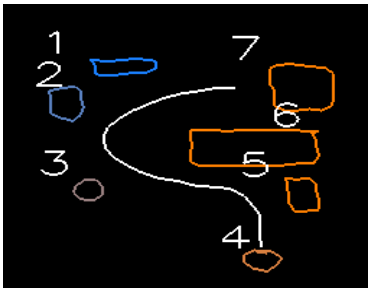
SET 1: In these sketches the user had not drawn one or more of the objects in the scene.



(k)



(l)



(m)

Figure A.2. Sketches of scene A with unequal number of objects

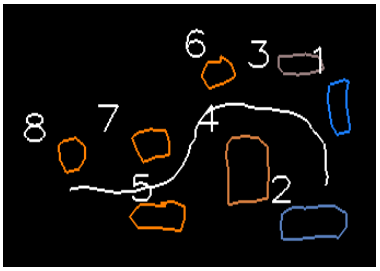
Table A.1. Scene A EASM Missing Objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.899	0.769	5	Object 6 (Sketch) -> Object 7' (Map)
l	0.912	0.81	7	
m	0.885	0.739	7	

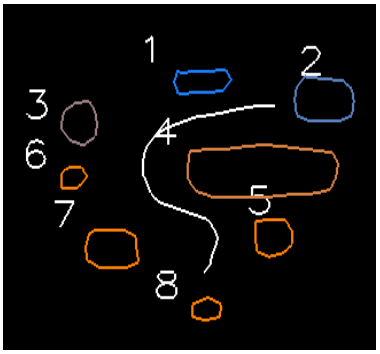
SET 2: To get a better understanding of how well the algorithm performs with missing objects, the sketches in which the user had drawn all the objects were also used. In these experiments, an object was removed from each sketch. The algorithm was run and the results were tabulated. The object was then inserted back and the next object was removed. The test was repeated until each of the objects had been removed.

Table A.2. Missing objects in various sketches of scene A

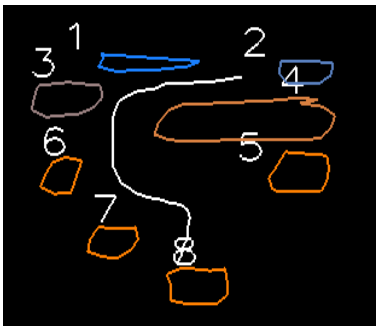
Scene A Sketch (b) – 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.9	0.795	7	
2	0.885	0.767	7	
3	0.89	0.759	7	
4	0.879	0.744	7	
5	0.887	0.754	7	
6	0.893	0.767	7	
7	0.89	0.795	6	Object 8 (Sketch) -> Object 7'(Map)
8	0.884	0.758	7	



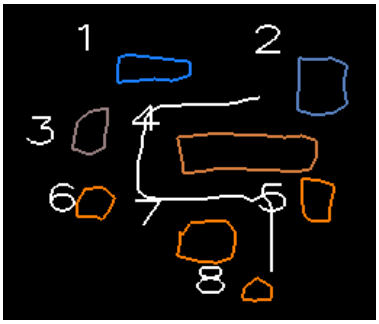
Scene A Sketch (c) – 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.889	0.772	7	
2	0.892	0.774	6	Object 1 (Sketch) -> Object 2'(Map)
3	0.872	0.745	5	Object 7,8 (Sketch) -> Object 6',7' (Map)
4	0.896	0.853	7	
5	0.872	0.793	7	
6	0.879	0.742	7	
7	0.867	0.757	6	Object 8 (Sketch) -> Object 7'(Map)
8	0.879	0.777	7	



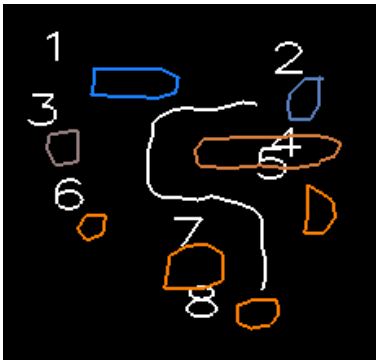
Scene A Sketch (d) – 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.864	0.723	7	
2	0.867	0.725	6	Object 1 (Sketch) -> Object 2' (Map)
3	0.873	0.753	6	
4	0.902	0.827	7	
5	0.849	0.712	7	
6	0.863	0.742	7	
7	0.861	0.743	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.865	0.746	7	



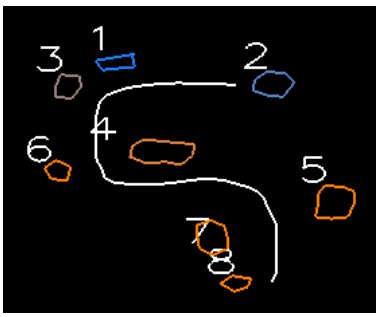
Scene A Sketch (e) – 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.912	0.792	7
2	0.902	0.769	7
3	0.916	0.793	7
4	0.908	0.79	7
5	0.906	0.776	7
6	0.909	0.765	7
7	0.905	0.772	7
8	0.909	0.798	7



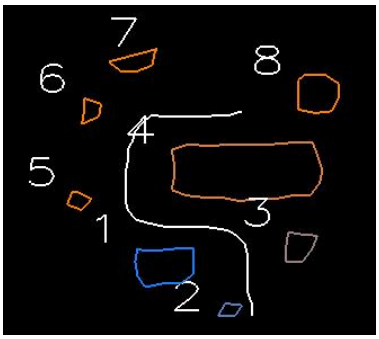
Scene A Sketch (f) – 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.907	0.802	7	
2	0.912	0.78	7	
3	0.89	0.765	7	
4	0.884	0.787	7	
5	0.867	0.754	7	
6	0.9	0.792	7	
7	0.892	0.786	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.889	0.764	7	



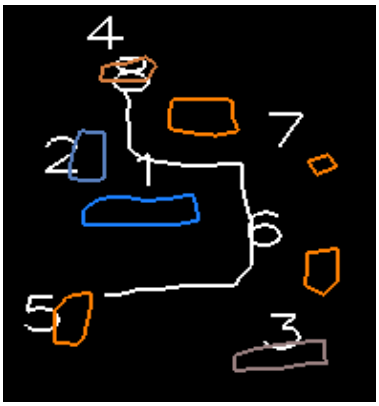
Scene A Sketch (g) – 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.915	0.9	7	
2	0.908	0.898	6	Object 1 (Sketch) -> Object 2' (Map)
3	0.913	0.901	7	
4	0.927	0.903	7	
5	0.917	0.912	7	
6	0.903	0.892	7	
7	0.916	0.882	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.906	0.906	7	



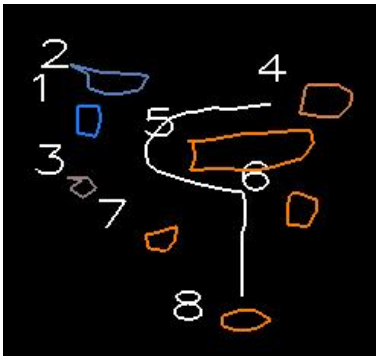
Scene A Sketch (h) – 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.914	0.835	7
2	0.913	0.831	7
3	0.9	0.801	7
4	0.924	0.869	7
5	0.906	0.821	7
6	0.894	0.785	7
7	0.899	0.793	7
8	0.9	0.806	7



Scene A Sketch (i) – 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.894	0.766	7	
2	0.907	0.795	7	
3	0.903	0.802	7	
4	0.903	0.77	7	
5	0.893	0.751	7	
6	0.894	0.768	7	
7	0.896	0.759	7	
8	0.9	0.795	6	Object 4 (Sketch) -> Object 7' (Map)



Scene A Sketch (j) – 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.919	0.842	7
2	0.929	0.86	7
3	0.917	0.82	7
4	0.913	0.822	7
5	0.935	0.903	7
6	0.918	0.842	7
7	0.912	0.829	7
8	0.917	0.842	7



APPENDIX B

Scene matching between scene B (physical map) and sketches with unequal number of objects

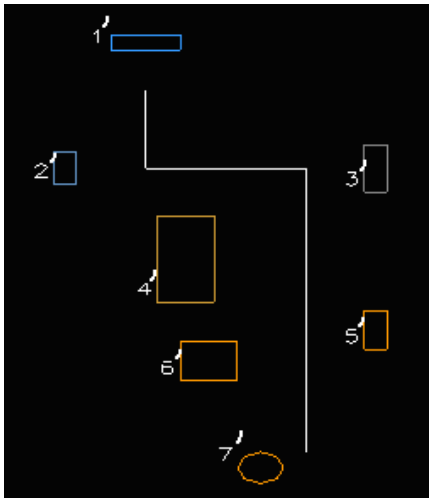
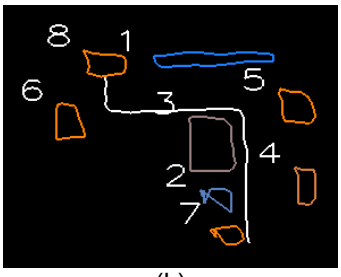
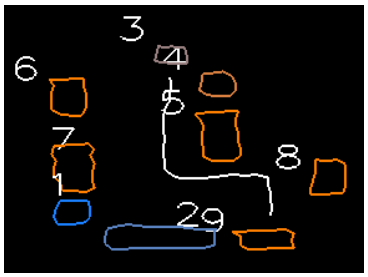


Figure B.1. Physical map for scene B

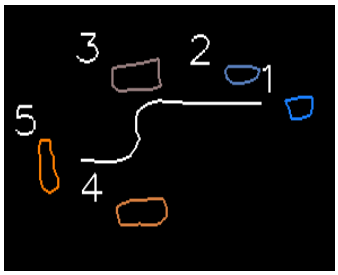
SET 1: In these sketches the user had not drawn one or more of the objects or had drawn extra objects.



(k)



(l)



(m)

Figure B.2. Sketches for scene B with unequal number of objects

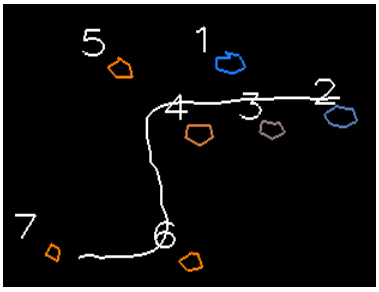
Table B.1. Scene B EASM Missing Objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.899	0.755	7	
l	0.901	0.782	6	Object 7 (Sketch) -> Object 3' (Map)
m	0.902	0.788	4	Object 1 (Sketch) -> Object 6' (Map)

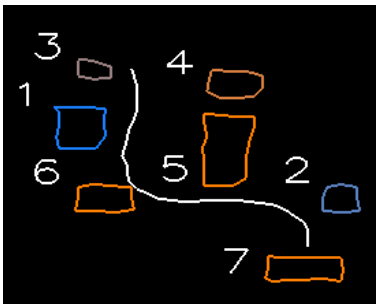
SET 2: To get a better understanding of how well the algorithm performs with missing objects, the sketches in which the user had drawn all the objects were also used. In these experiments, an object was removed from each sketch. The algorithm was run and the results were tabulated. The object was then inserted back and the next object was removed. The test was repeated until each of the objects had been removed.

Table B.2. Missing objects in various sketches of scene B

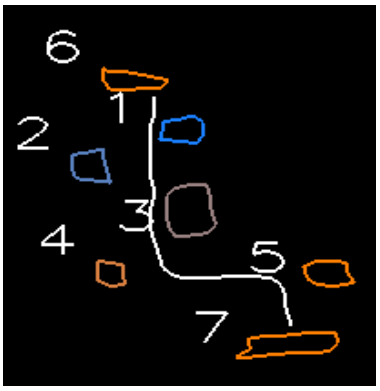
Scene B Sketch (b) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.906	0.823	6
2	0.917	0.874	6
3	0.907	0.87	6
4	0.921	0.854	6
5	0.92	0.86	6
6	0.902	0.816	6
7	0.872	0.786	6



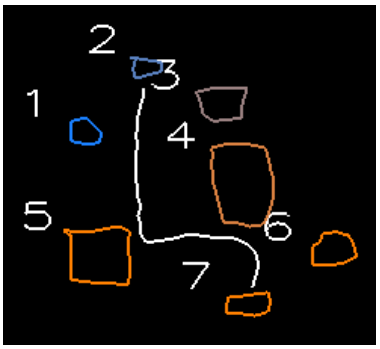
Scene B Sketch (c) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.917	0.847	5	Object 3 (Sketch) -> Object 5' (Map)
2	0.873	0.767	6	
3	0.905	0.824	6	
4	0.886	0.813	6	
5	0.882	0.826	6	
6	0.924	0.855	6	
7	0.918	0.873	6	



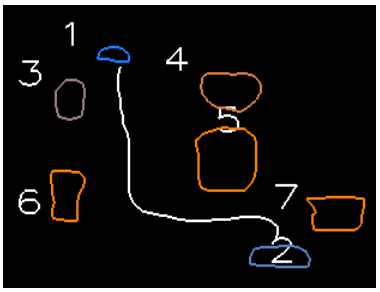
Scene B Sketch (d) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.913	0.856	5	Object 6 (Sketch) -> Object 6' (Map)
2	0.924	0.843	6	
3	0.919	0.81	6	
4	0.913	0.812	6	
5	0.918	0.83	6	
6	0.925	0.856	6	
7	0.94	0.887	6	



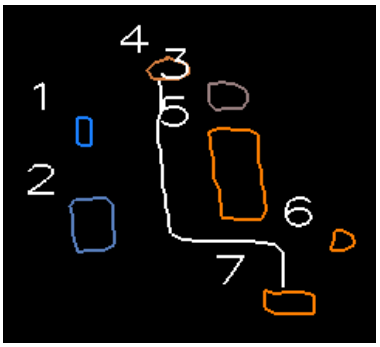
Scene B Sketch (e) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.912	0.773	6
2	0.913	0.801	6
3	0.907	0.771	6
4	0.913	0.798	6
5	0.914	0.809	6
6	0.908	0.788	6
7	0.909	0.789	6



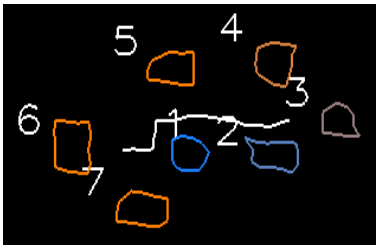
Scene B Sketch (f) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.891	0.789	6	
2	0.881	0.802	6	
3	0.903	0.82	6	
4	0.879	0.771	5	Object 1 (Sketch) -> Object 6' (Map)
5	0.882	0.83	6	
6	0.88	0.789	6	
7	0.881	0.799	6	



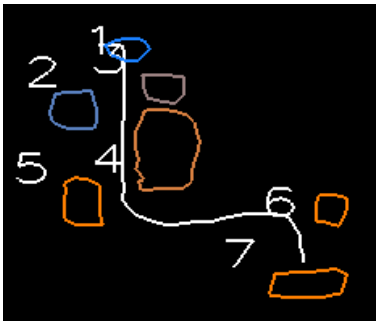
Scene B Sketch (g) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.92	0.821	6	
2	0.923	0.839	6	
3	0.908	0.798	5	Object 4 (Sketch) -> Object 6' (Map)
4	0.908	0.789	6	
5	0.912	0.803	6	
6	0.909	0.805	6	
7	0.909	0.815	6	



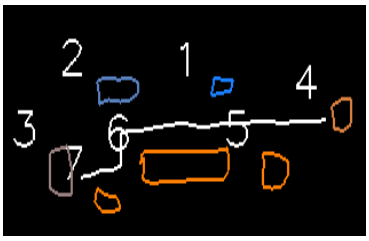
Scene B Sketch (h) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.888	0.727	6	
2	0.879	0.794	5	Object 3 (Sketch) -> Object 6' (Map)
3	0.889	0.787	6	
4	0.894	0.767	6	
5	0.891	0.762	6	
6	0.892	0.805	6	
7	0.875	0.73	6	



Scene B Sketch (i) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.904	0.819	6	
2	0.92	0.841	6	
3	0.91	0.836	5	Object 1 (Sketch) -> Object 6' (Map)
4	0.917	0.82	6	
5	0.906	0.824	6	
6	0.905	0.809	6	
7	0.902	0.791	6	

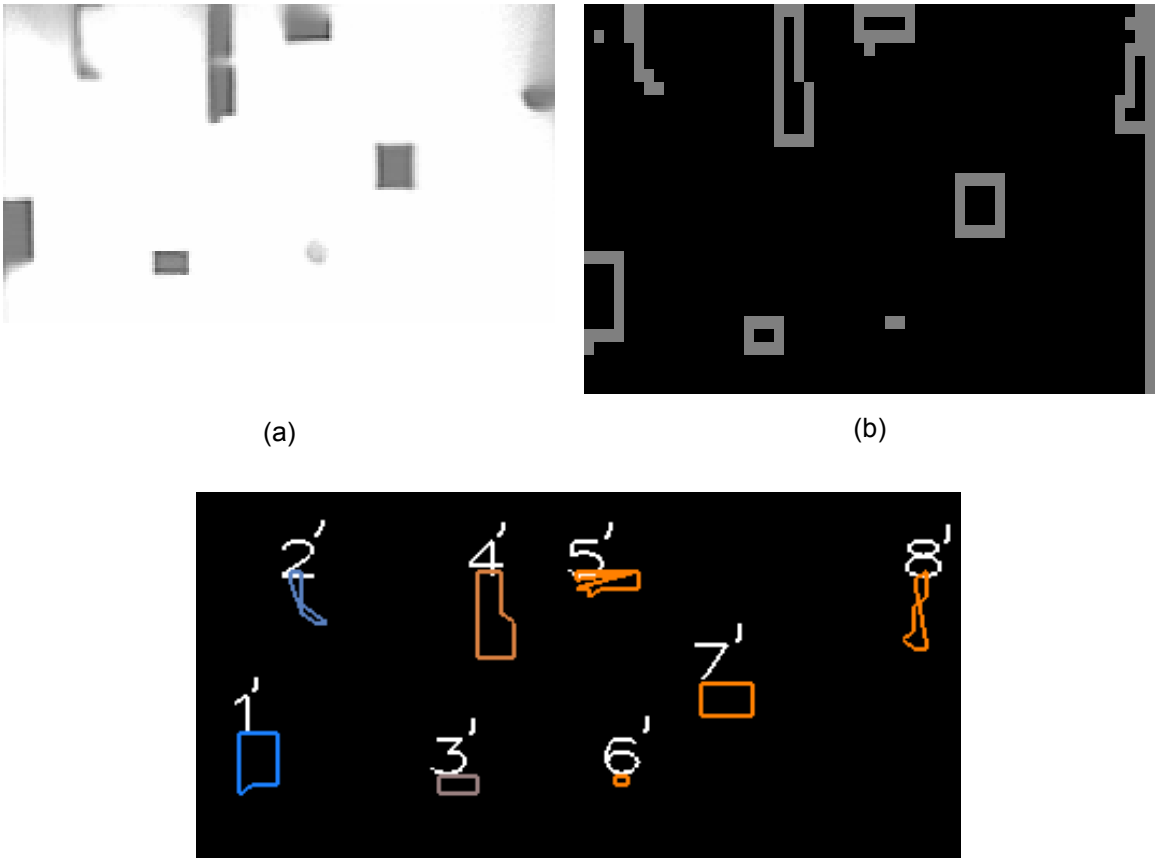


Scene B Sketch (j) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.929	0.852	6	
2	0.914	0.806	6	
3	0.928	0.846	6	
4	0.907	0.774	6	
5	0.907	0.772	5	Object 4 (Sketch) -> Object 6' (Map)
6	0.921	0.822	6	
7	0.91	0.778	6	



APPENDIX C

Scene matching between scene A (OGM) and sketches with unequal number of objects

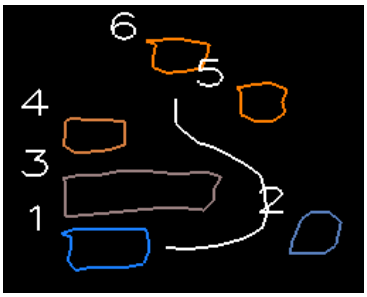


(a) (b)

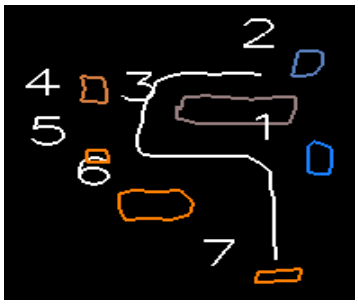
(c)

Figure C.1. (a) OGM built by the robot, (b) OGM after image processing operations, (c) OGM with labeled objects (labels not used for performing scene matching) .

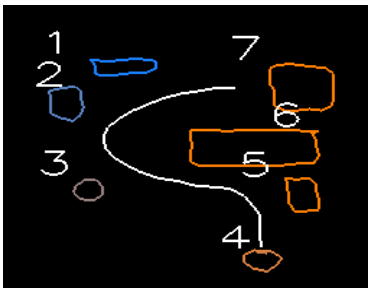
SET 1: In these sketches the user had not drawn one or more of the objects in the scene.



(k)



(l)



(m)

Figure C.2. Sketches of scene A with unequal number of objects

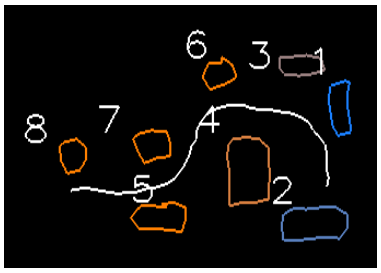
Table C.1. Scene A EASM Missing Objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.88	0.731	6	
l	0.912	0.81	7	
m	0.866	0.745	6	Object 3 (Sketch) -> Object 7' (Map)

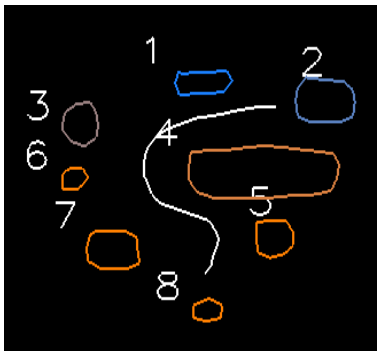
SET 2: To get a better understanding of how well the algorithm performs with missing objects, the sketches in which the user had drawn all the objects were also used. In these experiments, an object was removed from each sketch. The algorithm was run and the results were tabulated. The object was then inserted back and the next object was removed. The test was repeated until each of the objects had been removed.

Table C.2. Missing objects in various sketches of scene A

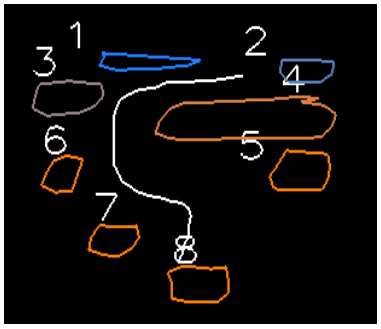
Scene A Sketch (b) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.881	0.73	7	
2	0.866	0.732	7	
3	0.876	0.735	7	
4	0.86	0.709	7	
5	0.861	0.7	7	
6	0.877	0.749	7	
7	0.859	0.74	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.85	0.672	7	



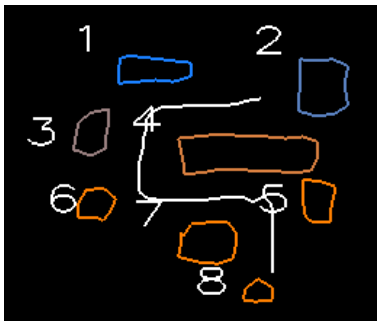
Scene A Sketch (c) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.887	0.742	7	
2	0.877	0.738	6	Object 1 (Sketch) -> Object 2' (Map)
3	0.864	0.676	7	
4	0.873	0.74	7	
5	0.864	0.683	7	
6	0.868	0.702	7	
7	0.867	0.683	7	
8	0.87	0.696	7	



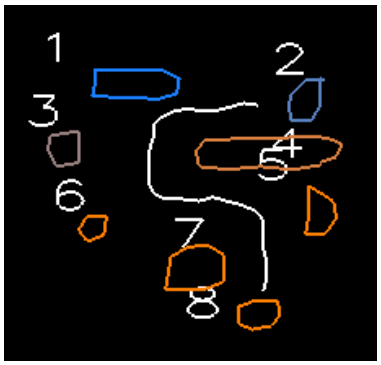
Scene A Sketch (d) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.868	0.707	7	
2	0.835	0.656	6	Object 1 (Sketch)-> Object 2' (Map)
3	0.869	0.713	7	
4	0.854	0.721	7	
5	0.844	0.666	7	
6	0.852	0.691	7	
7	0.833	0.648	6	Object 8 (Sketch)-> Object 7' (Map)
8	0.855	0.712	7	



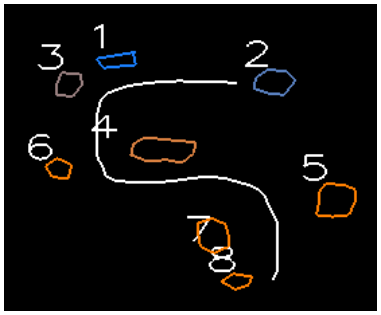
Scene A Sketch (e) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.892	0.754	7
2	0.879	0.725	7
3	0.88	0.744	7
4	0.882	0.732	7
5	0.882	0.708	7
6	0.889	0.738	7
7	0.878	0.712	7
8	0.879	0.702	7



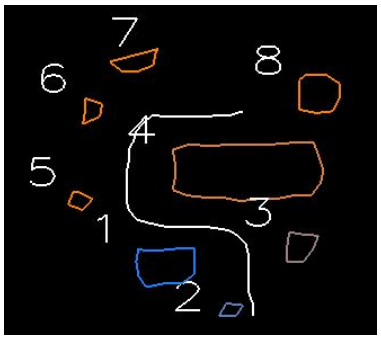
Scene A Sketch (f) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.88	0.782	7
2	0.871	0.724	7
3	0.869	0.701	7
4	0.878	0.742	7
5	0.871	0.696	7
6	0.881	0.754	7
7	0.877	0.724	7
8	0.871	0.72	7



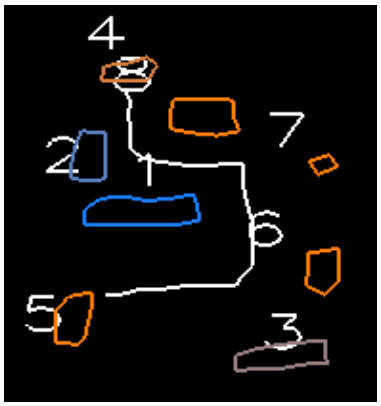
Scene A Sketch (g) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.885	0.802	7	
2	0.88	0.792	7	
3	0.882	0.797	7	
4	0.897	0.801	7	
5	0.871	0.776	7	
6	0.865	0.754	7	
7	0.872	0.731	6	Object 8 (Sketch) -> Object 7' (Map)
8	0.881	0.761	7	



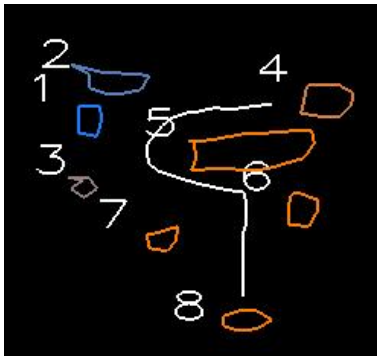
Scene A Sketch (h) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.882	0.735	7
2	0.885	0.736	7
3	0.875	0.715	7
4	0.89	0.773	7
5	0.877	0.731	7
6	0.876	0.722	7
7	0.875	0.729	7
8	0.864	0.701	7



Scene A Sketch (i) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.89	0.742	7	
2	0.893	0.747	7	
3	0.898	0.773	7	
4	0.888	0.734	7	
5	0.892	0.752	7	
6	0.884	0.729	7	
7	0.898	0.775	7	
8	0.892	0.749	6	Object 4 (Sketch) -> Object 7' (Map)

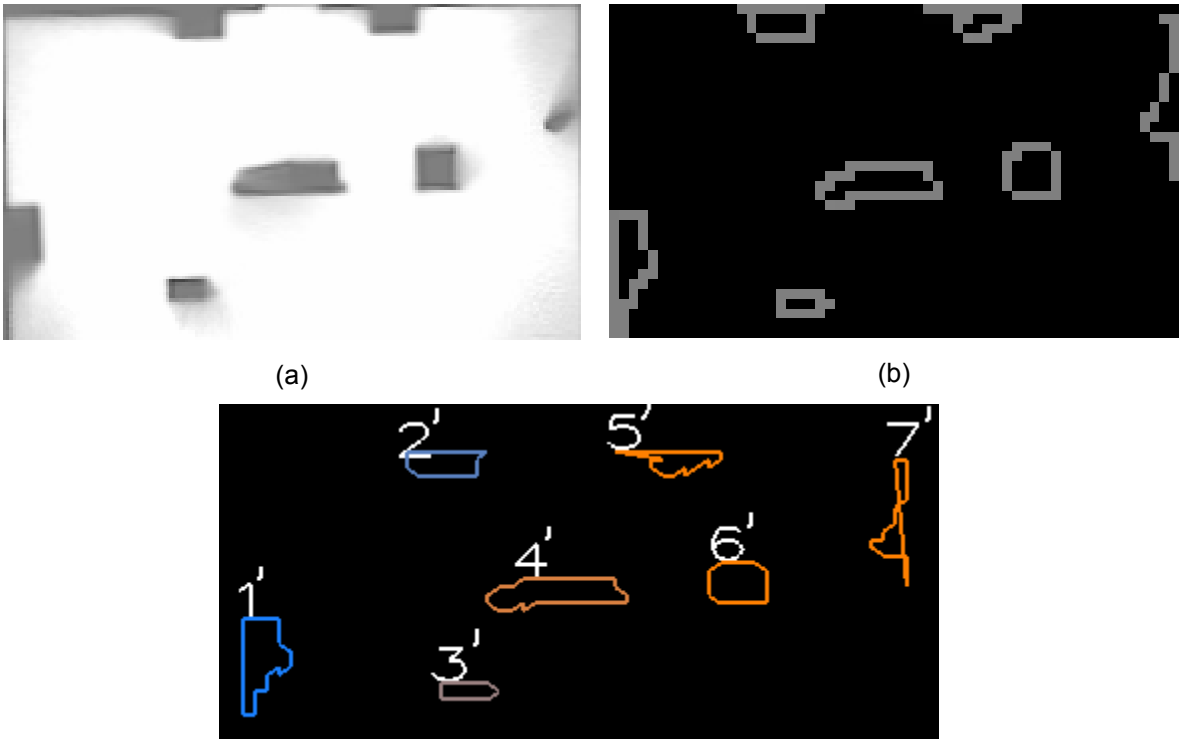


Scene A Sketch (j) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.895	0.783	6	Object 3 (Sketch) -> Object 6' (Map)
2	0.899	0.781	7	
3	0.885	0.76	7	
4	0.881	0.739	7	
5	0.894	0.784	7	
6	0.883	0.744	7	
7	0.88	0.745	7	
8	0.876	0.732	7	



APPENDIX D

Scene matching between scene B (physical map) and sketches with unequal number of objects

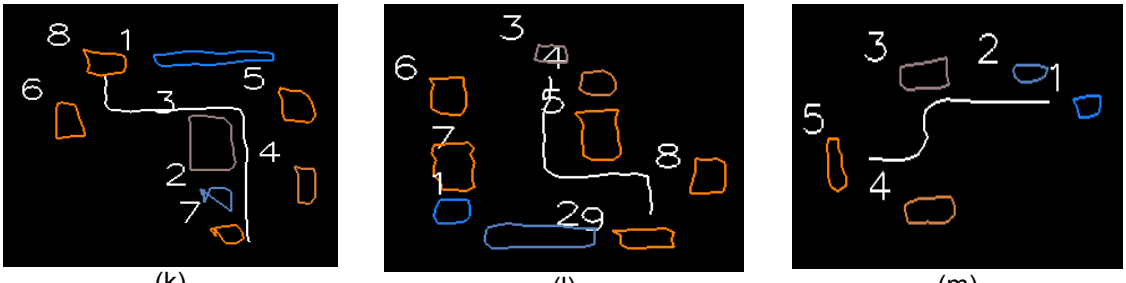


(a) (b)

(c)

Figure D.1. (a) OGM built by the robot, (b) OGM after image processing operations, (c) OGM with labeled objects (labels not used for performing scene matching) .

SET 1: In these sketches the user had not drawn one or more of the objects or had drawn extra objects.



(k) (l) (m)

Figure D.2. Sketches for scene B with unequal number of objects

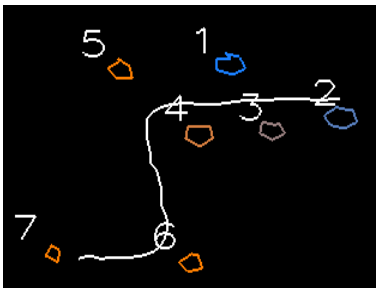
Table D.1. Scene B EASM Missing Objects

Sketch	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
k	0.916	0.826	7	
l	0.922	0.844	5	Object 7 (Sketch) -> Object 2' (Map)
m	0.902	0.788	3	Object 2, 3 (Sketch) -> Object 6', 4'(Map)

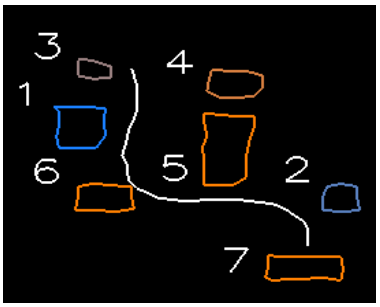
SET 2: To get a better understanding of how well the algorithm performs with missing objects, the sketches in which the user had drawn all the objects were also used. In these experiments, an object was removed from each sketch. The algorithm was run and the results were tabulated. The object was then inserted back and the next object was removed. The test was repeated until each of the objects had been removed.

Table D.2. Missing objects in various sketches of scene B

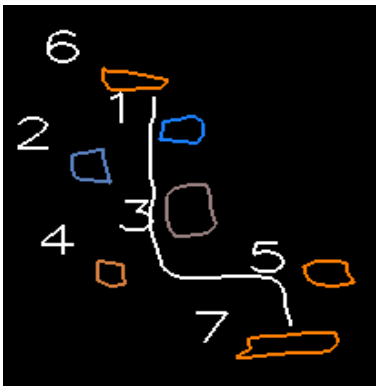
Scene B Sketch (b) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.874	0.68	6	
2	0.885	0.738	6	
3	0.874	0.676	5	Object 2 (Sketch) -> Object 6' (Map)
4	0.886	0.712	6	
5	0.892	0.758	6	
6	0.894	0.746	6	
7	0.897	0.769	6	



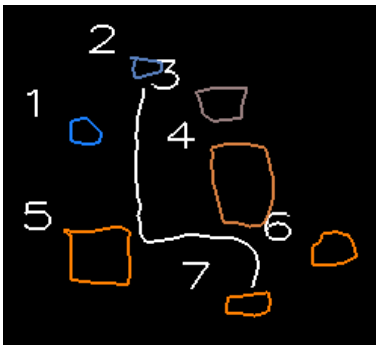
Scene B Sketch (c) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.922	0.823	5	Object 3 (Sketch) -> Object 5' (Map)
2	0.901	0.83	6	
3	0.904	0.811	6	
4	0.9	0.796	6	
5	0.907	0.828	6	
6	0.918	0.825	4	Object 1, 3 (Sketch) -> Object 3', 5' (Map)
7	0.892	0.805	6	



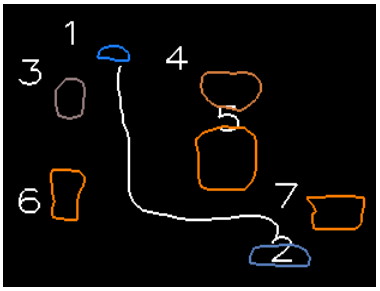
Scene B Sketch (d) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.922	0.876	5	Object 6 (Sketch) -> Object 5' (Map)
2	0.932	0.868	6	
3	0.94	0.89	6	
4	0.942	0.893	6	
5	0.932	0.888	6	
6	0.918	0.856	6	
7	0.935	0.898	6	



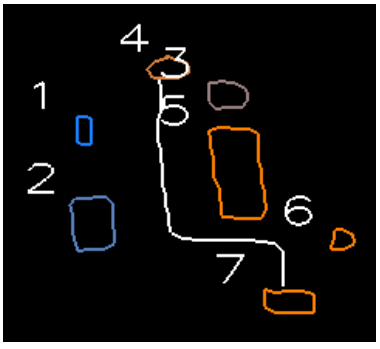
Scene B Sketch (e) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.911	0.823	6
2	0.881	0.83	6
3	0.914	0.811	6
4	0.932	0.796	6
5	0.907	0.828	6
6	0.912	0.825	6
7	0.932	0.805	6



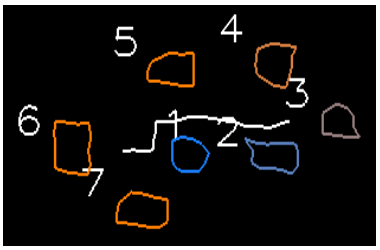
Scene B Sketch (f) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.91	0.792	6
2	0.891	0.804	6
3	0.903	0.83	6
4	0.922	0.812	6
5	0.905	0.83	6
6	0.902	0.789	6
7	0.934	0.825	6



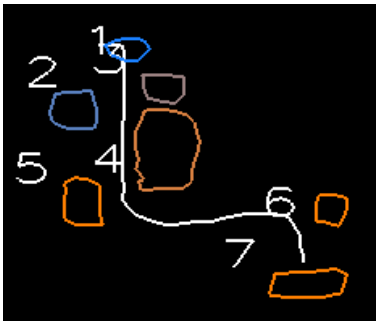
Scene B Sketch (g) - 1 Missing Object			
Missing Object	Confidence values for		Correct Matches
	FMAP(ζ)	OMAP(Ω)	
1	0.922	0.846	6
2	0.931	0.886	6
3	0.906	0.778	6
4	0.921	0.814	6
5	0.919	0.831	6
6	0.921	0.8	6
7	0.906	0.792	6



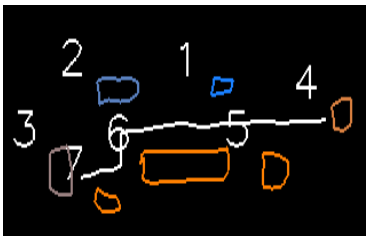
Scene B Sketch (h) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.908	0.742	6	
2	0.892	0.774	5	Object 3 (Sketch) -> Object 6' (Map)
3	0.877	0.781	6	
4	0.884	0.81	6	
5	0.901	0.792	6	
6	0.89	0.825	6	
7	0.876	0.766	6	



Scene B Sketch (i) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.916	0.824	6	
2	0.928	0.817	6	
3	0.902	0.802	5	Object 1 (Sketch) -> Object 6' (Map)
4	0.904	0.797	6	
5	0.911	0.81	6	
6	0.912	0.792	6	
7	0.892	0.804	6	



Scene B Sketch (j) - 1 Missing Object				
Missing Object	Confidence values for		Correct Matches	Incorrect Matches
	FMAP(ζ)	OMAP(Ω)		
1	0.929	0.852	6	
2	0.914	0.806	6	
3	0.928	0.846	6	
4	0.907	0.774	6	
5	0.907	0.772	5	Object 4 (Sketch) -> Object 6' (Map)
6	0.921	0.822	6	
7	0.91	0.778	6	



REFERENCES

[Aschwanden and Guggenbuhl, 1992] P. Aschwanden and W. Guggenbuhl, “Experimental results from a comparative study on correlation-type registration algorithms,” in *Robust Computer Vision*, pp. 268-289, 1992.

[Bailey, 2003] 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.

[Boland et. al., 1980] 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.

[Chakravarty, 1981] I. Chakravarty, “A single-pass chain generating algorithm for region boundaries,” *Computer Graphics and Image Processing*, vol. 15, 1981, pp. 182-193.

[Chronis and Skubic, 2004] 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.

[Fong, Thorpe, and Baur, 2003] T. Fong, C. Thorpe, C. Baur, “Multi-robot remote driving with collaborative control,” *IEEE Trans. on Industrial Electronics*, vol. 50, no. 4, Aug 2003, pp. 699-704.

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

[Holland, 1975] J. H. Holland, “Adaptation in natural and artificial life,” Ann Arbor, MI: University of Michigan Press, 1975.

[Kawamura et. al., 2003] K. Kawamura, A. B. Koku, D. M. Wilkes, R. A.

Peters II, and A. Sekmen, “Toward perception based navigation using Egospheres,” *Int’l Journal of Robotics and Automation*, Aug 2003.

[Keller and Szatendra, 1990] J. M. Keller and L. Szatendra, “Spatial relations among fuzzy subsets of an image,” *Proc. Of the First Int’l Symposium on Uncertainty Modeling and Analysis,* College Park, Univ of Maryland, pp. 207-211, 1990.

[Keller, Gader, Wang, 1999] 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.

[Kennedy and Eberhart, 1995] J. Kennedy, R. Eberhart, “Particle Swarm Optimization,” *IEEE Int’l Conf. On Neural Networks*, vol. 4, pp. 1942-1948, Nov, 1995.

[Krishnapuram et. al., 1993] 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.

[Lin et. al., 2001] C. Y. Lin, M. Wu, J. A. Bloom, I.J. Cox, M.L. Miller, and Y. M. Lui, “Rotation, scale and translation resilient watermarking for images,” in *IEEE Trans. on Image Proc*., vol. 10, no. 5, pp. 767-782, May 2001.

[Lundberg et. al., 2003] C. Lundberg, C. Barck-Holst, J. Folkeson, H. Christensen, “PDA interface for a field robot”, *IEEE/RSJ Int’l Conf. on Robotics and Systems*, Las Vegas, Nevada, Oct 2003, pp. 2882-2888.

[Mahoney et. al., 2002] J. V. Mahoney and M. P. J. Fromherz, “Three main concerns in sketch recognition and an approach to addressing them,” *2002 AAAI Spring Symposium, Sketch understanding workshop*, Stanford Univ., AAAI technical report SS-02-08, pp. 105-112.

[Matsakis and Wendling, 1999], 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.

[Matsakis et. al., 2004] 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.

[Miyajima and Ralescu, 1994] 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.

[Parekh et. al., 2007] G. Parekh, M. Skubic, O. Sjahputera and J. M. Keller, "Scene matching between a map and a hand drawn sketch using spatial relations," *IEEE Conf. on Robotics and Automation*, Rome, Italy, April, 2007.

[Perzanowski et. al., 2001] D. Perzanowski, A. Shultz, W. Adams, E. Marsh and M. Bugajska, "Building a multimodal human robot interface," *IEEE Intelligent Systems,* Jan 2001, pp. 16-21.

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

[Rosenfeld, 1979] A. Rosenfeld, "Fuzzy Digital Topology," *Information Control,* vol. 40, 1979, pp. 76-87.

[Rosenfeld, 1983] A. Rosenfeld, "On connectivity properties of Grayscale Pictures," *Pattern Recognition,* vol. 16, 1983, pp. 47-50.

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

[Shi et. al., 1997] 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.

[Sjahputera et. al., 2000] 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.

[Sjahputera et. al., 2003] 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.

[Sjahputera, 2004] 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.

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

[Sjahputera and Keller, 2005] 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.

[Sjahputera and Keller, 2006] 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.

[Skubic et. al., 2003] 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.

[Skubic et. al., 2004] 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.

[Skubic et. al., 2006] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz, "Using a Qualitative Sketch to Control a Team of Robots," in *Proceedings of the IEEE 2006 Intl. Conf. on Robotics and Automation,* Orlando, FL, May, 2006.

[Skubic et. al., 2007] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz, "Using a Hand Drawn Sketch to Control a Team of Robots," in *Autonomous Robot*, vol. 22, pp. 399-410, Jan 2007.

[Wang et. al., 1995] 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.

[Winston, 1975] P. Winston, “Learning structural descriptions from examples,” in P. Winston (ed.), *The Psychology of Computer Vision,* McGraw-Hill, New York, 1975.

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

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