I Spatially Integrate Therefore I Am… Lost? A New Benchmark For Autonomous Mobile Robot Navigation

Simon J Egerton, Vic Callaghan
University Of Essex, Wivenhoe Park, Colchester, CO4 3SQ, England
egers@essex.ac.uk vic@essex.ac.uk

Abstract

In this paper we introduce a spatial integration model that is currently under development. We have developed a set of tools and methods to help benchmark applications of the model to autonomous mobile robots situated in indoor and outdoor environments. We present the Lost Metric benchmark, a new evaluation method that is grounded to a pragmatic navigation task, localisation. We report on results of its application to some implementations of the model.

1. Introduction

With the growing number of spatial integration models being developed for autonomous mobile robots, there is also a growing need for a standard set of benchmarking tools with which to measure performance and to make comparisons, quantitatively. In a step towards this goal, we offer in this paper a metric that can be used in such a way. The benchmark we introduce is based upon a pragmatic navigation task that is central too most spatial models, localisation.

To demonstrate the "*Lost Metric*" benchmark, we use examples of the spatial integration model shown in Figure 1, which is under continual development.

In the remainder of the paper we give an overview of the spatial model we are developing and describe the *lost metric* and the benchmark. We report on the methods of evaluation and results of an application of the benchmark over a range of implementations of the model.

2. A Biologically Inspired Spatial Model

Figure 1 illustrates the overall model. The model introduces two concepts, firstly the notion of a "Perception Space" and secondly the notion of a "Geometric Space". The "Perception Space" is constructed directly from the robots sensory impressions. These sensor impressions are defined as sensor element activation's relating to some fixed physical location in the robots environment (generally not directly equitable to objects in the "human" perception domain) and we term these "Perception Signatures". These Perception Signatures are categorised by the "Perception Space" modules, each category formed is termed a "Perception Class". The "Geometric Space" module is a geometric framework and its purpose is to relate geometric areas to perception classes in the perception space modules. This forms "Perception Areas", which are homogenous areas sharing the same perception class, as illustrated by Figure 1. A navigable map is formed in the later stages by adding the geometric spaces together; this can be a selective process, the selection based on quality of sensory data or quality of navigation paths, for example. The inspiration for this model was originally derived from the biological literature and is based on mammalian spatial integration theory [1, 3]. This is a very brief overview of the model and [4] gives a fuller description of the model.

2.1. Model Benchmarking

With respects to the many varied implementations of the spatial model in the mammalian world, there are as many varied implementations in the autonomous mobile robot world too. Many factors effect how the model is applied to a robot situated in a real environment. Those factors primarily include the type of target environment, the abilities of the target robot, and the target application. It becomes apparent that we need to have some method of benchmarking each application of the model. Primarily, the benchmark should indicate which implementation within a given situation is the "better", the notion of "better" depending upon design criteria. And, while we believe that there is no generalised solution for the model, as the potential set of implementations over a range of environments, sensors and robots are as diametrically varied as they are numerous. Benchmarking might allow generalisations to be made over a range of environments, sensors, and robots.

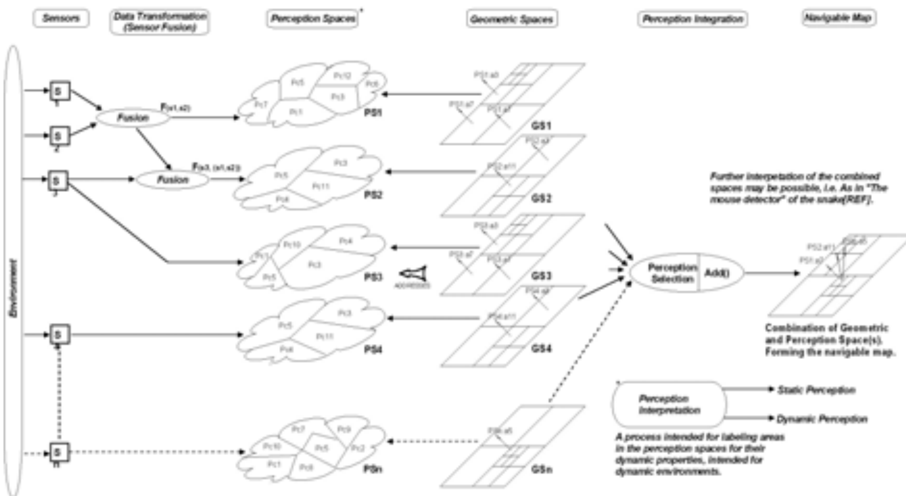

Figure 1. Architecture of the spatial integration model.

3. The *lost metric*

Generally, for purposes of navigation, a map is only useful if we know where we are within it. Furthermore, a map is really useful if we can use it to find where we are within it, if for some reason, we become lost. Moreover, a map that lets one localise, when lost, with the least amount of effort is more useful than one that does not. The *lost metric* is based upon this notion of usefulness, which then forms the basis of the benchmark to be described later. To measure the usefulness of a generated map in these terms we consider random locations within the mapped area and calculate the amount of effort required to locate ones self again from these locations. The amount of effort this takes can be quantified into the number of steps one takes, and also the size and similarity of the areas with the map that made the steps necessary in the first instance.

These notions are reflected in the design of the *Lost Metric* for which we introduce two new concepts, firstly the notion of an "Adaptive Perceptual Area Inference Map" and secondly the "*Lost Metric* Function" itself, now described in the following sections.

3.1. Adaptive perceptual area inference map

The first three stages of the model are arguably the most important ashey deal with the robot's sensors, the construction of the perception signatures, and the categorisation of those signatures. Also an implementation of these stages involves many design decisions, these decisions influenced by the application of the robot, as discussed above. In contrast, the later three stages are common to all implementations; they contain the components that are theorised to be common to all mammalian examples of the biological model [3]. For these reasons we apply the *lost metric* benchmark to the third stage, the Perception Space Modules, to quantitatively evaluate the design decisions taken. We need to construct a perceptual map to facilitate this evaluation. The perceptual map exhibits the properties needed for the *lost metric* evaluation, as explained in following sections. The perceptual map grounds the perceptual classes within a perception space into a geometric frame, assigning them to one or more Perception Areas.

A perceptual map is calculated from the perceptual classes of a perception space module and their geometric locations, as provided by odometry data, for the purposes of evaluation. The odometry from the Pioneer robot is used directly; there is no need for filtering, since the data is sufficiently accurate from within the evaluation environments in the laboratory.

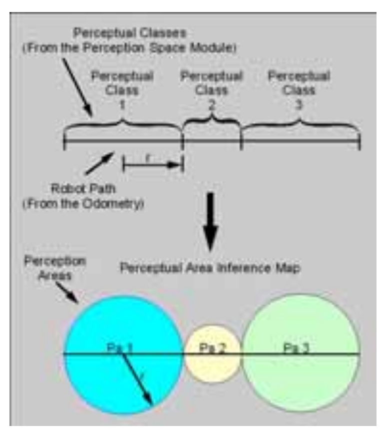

Figure 2. From the perception signatures experienced by the robot along its path, Perceptual Area Inference Maps are generated using the robot's odometry and the perceptual classes generated from the perception space modules.

The odometry is used to assign perception classes along the robots path were they were experienced. Each location along the path is grown and or shrunk, so that no location of a different perception class overlaps. Locations that overlap and are of the same perception class are merged and a Perception Area formed. Where the situation arises such that an area is of the same type, but do not overlap, they are treated as separate perception areas. Such a situation may arise from perceptual aliasing or from two perceptual classes being in close proximity along a stretch of the robot's path. This process is adaptive and converges on a set of Perception Areas that "best" fit the parts of the environment the robot has explored. The area for each Perception Area is calculated. Areas are considered as neighbouring if they touch or if they have been neighbours along the path of the robot. These notions are illustrated in Figure 2; perception areas are assigned colours relating to their perceptual class. When the signature classes are grown from their locations in this way, it speculatively extends their region of influence away from the original path over an area as yet unexplored by the robot. Hence, the term "*inference maps*". Speculation is required since the model is grounded in perceptions that are experienced at locations. The speculation is based on the continuity of successive signature classes along the path of the robot, Figure 2.

3.2. *Lost metric* function

The *lost metric* can be applied to any model that models the environment with the two properties of having defined areas of similarity and of having relations between those areas. In the case where the model or part of the model does not explicitly have these properties then an inference map maybe generated, similar to the one described above. We use the perceptual inference map to evaluate the usefulness of the perception space modules with the *lost metric*, and we use the inference map as an example to explain the *lost metric* in the following.

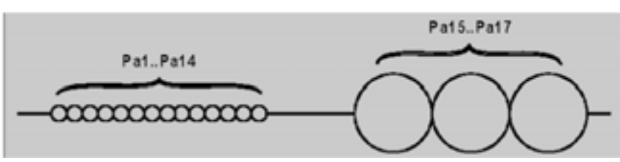

Figure 3. A line of the robot's path is described by a set Perception Areas. This description can be at two extremes. The first extreme shows the path described by many small perception-areas Pa1 to Pa14. A perception area may reduce to a point at its most extreme. The other extreme shows the path described by few large perceptions-areas Pa15 to Pa17. A perception area may increase to fill the entire environment at its most extreme.

If the robot is lost within the perceptual inference map, and if it occupies a perception area that is unique, no localisation is necessary. However, if the area is not unique, then it is necessary to visit other perception areas until a unique perception area, or a unique sequence of perception areas, is encountered. Therefore, a localisation path within the map may contain one or more perception areas and one or more localisation steps. A localisation step is taken to be the number of perception areas with the localisation path. The key to the *lost metric* is the interpretation of how a length of a robot's path can be described in the inference map by a set of perceptual areas; the two extremes of the description are illustrated in Figure 3. At one extreme, numerous small perception areas describe the path and at the other extreme, the path is described by one large perception area. Moreover, either of these extremes maybe a sign that the classification in the Perception Space module has failed. However, it maybe that the application calls for such extremes and is actually desirable. At the extreme of many small perception areas, storage and computational costs increase, as do the risks of perceptual aliasing. However, it is desirable to be able to localise within the space of one perception area.

Two values are calculated for each localisation path, a "Localisation Steps Value" and a "Localisation Area" value, and these are used to calculate a *lost metric* value for the path.

The "Localisation Steps" value is equal to the number of perception areas in a localisation path; the value can range from one upwards to the number of perception areas in the inference map. The step value is passed to the "Localisation Steps" function defined by equation (1). The function returns a value in the range of 0 and 1. The function's behaviour depends on the values of two parameters. These parameters form part of the *lost metric* criteria. The function returns 1 in the best case, the criteria have been met, and 0 in the worst case. The worst case also accounts for the situations where the localisation steps are infinite, if the robot were unable to localise, for example. The number of steps to be considered as ideal is set by t_θ and tolerance for steps beyond t_θ is set by the value of θ. The criteria becomes more relaxed for larger values of t_θ and θ. For example, if $t_\theta = 5$, then up to and including 5 localisation steps in a localisation path is considered as ideal. If $\theta = 0$ then no tolerance is given to values beyond 5 steps. Naturally, $t_\theta = 1$ and $\theta = 0$, are the ideal. However, what the values actually are pragmatically, depends on the conditions acceptable for the desired application.

$$f_{LocalisationSteps}(x) = \begin{cases} \theta^{x-t_\theta} & \text{if } x > t_\theta \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

The "Localisation Area" is equal to the sum of the perception area areas in a localisation path; the value can range from the area of a single location in the map, to the area of the entire explored environment. The area value is passed to the "Localisation Area" function defined by equation (2). The function takes the localisation area value as its argument and returns a value in the range of 0 and 1. The function's behaviour depends on the values of four parameters.

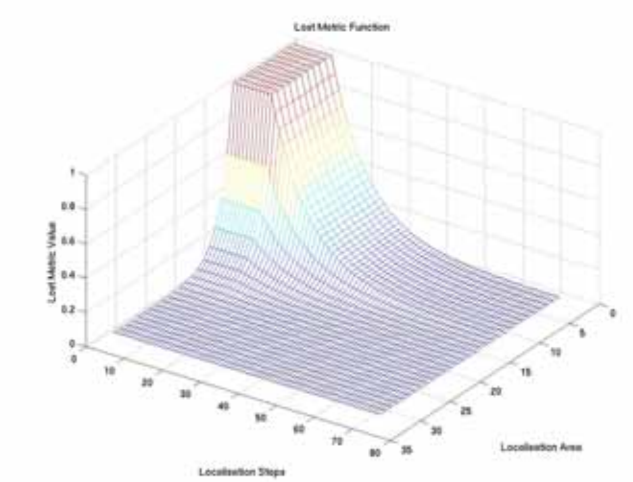

Figure 4. Graph of the *Lost Metric* function. The parameters are $\omega = 5.0$, $\lambda = 0.5$, $t_w = 1$, $t_\lambda = 10$, $\theta = 0.9$, $t_\theta = 10$, and were chosen for illustrative purposes. The plateau represents the ideal case.

These parameters form part of the *lost metric* criteria. The function returns 1 in the best case, the criteria have been met, and 0 in the worst case. The worst case also accounts for the situations where the localisation area is infinite, if the robot were unable to localise. The range of area considered ideal is set between the minimum t_ω and the maximum t_λ. The tolerance of areas below the ideal minimum is set by ω and the tolerance of areas above the ideal maximum is set by λ. The criteria becomes more relaxed for larger ranges of area, between t_ω and t_λ and for smaller values of ω and λ. For example, if $t_\omega = 0.3cm^2$ and $t_\lambda = 0.5cm^2$ then all areas between and including these sizes would be considered ideal. If $\omega = 0.5$, then reasonable tolerance is given to areas below 0.3cm^2 and if $\lambda = 5.0$, then very little tolerance is given for areas beyond 0.5cm^2. The amount of tolerance given to a criteria maybe visualised by plotting function graph. The ideal localisation area depends on the conditions acceptable for the desired application.

$$f_{LocalisationArea}(y) = \begin{cases} e^{-\omega(t_\omega - y)} & \text{if } y > t_\omega \\ e^{-\lambda(y - t_\lambda)} & \text{if } y < t_\lambda \\ 1 & \text{otherwise} \end{cases} \qquad (2)$$

The *Lost Metric* function is defined by equation (3) and is the product of equation (1) and equation (2). The function takes both the localisation steps and the localisation area values as its argument and returns a value in the range of 0 and 1. The function returns 1 in the best case localisation path and 0 in the worst case, according to the defined criteria. A graph of an instance of the *Lost Metric* function defined by equation (3) is illustrated in Figure 4.

$$f_{LostMetric}(x, y) = f_{LocalisationSteps}(x) \bullet f_{LocalisationArea}(y) \qquad (3)$$

Although we have illustrated the *Lost Metric* with the perceptual inference map, the notions are readily extendable to other models that exhibit the localisation steps and localisation area properties, as described above.

4. Paths of evaluation

If the robot is lost within a mapped area, then it needs to localise, using the map. The localisation process may follow two philosophies. When the robot knows that it is lost, it may choose to ask the following questions:

1) Where am I?

2) Where was I when I became lost?

There may be a natural order to these questions, once the first has been answered the robot is then in a position of answer the second. It may be argued that a true measure of how useful a map is, is one that considers the costs involved in answering both these questions. However, for the purposes of assessing the Perceptual Space module we choose to answer the later question, as it seems the more appropriate of the two.
In order to justify this view, we illustrate the differences between the two questions by considering them within a simple Starmaze environment, illustrated in Figure 5. The Starmaze is constructed from a set of perception areas. The centre of the Starmaze is of one perception type and the nine arms of the Starmaze are of the same perception type but differing from the centre.
We examine each of the localisation questions in turn with respects this environment, assuming the robot is lost somewhere inside. We further assume that the robot's odometry has malfunctioned and is unreliable. Therefore, it is using only its external perceptions to localise itself with, but it does have an accurate perceptual map of the environment.

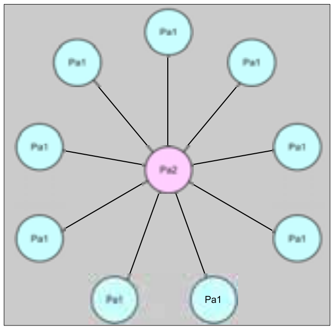

Figure 5. A Starmaze, all the areas of the maze are of the same perception type, with the exception of maze centre.

To answer the question of "Where Am I?" If the robot were not fortunate enough to be currently in a unique location, for example, in one of the maze arms, it would need to explore further. The robot would need to explore to either find a location that was unique, the maze centre in this case, or to explore a path that was unique to the subsequent current location. Under these conditions, the robot can reliably answer the question 100% of the time, by navigating to the centre of the maze. However, to answer the question of "Where was I when I became lost?" If the robot were again not currently in a unique location, one of the maze arms, then it would need to explore a path that was unique starting from the location of where it currently was. Under these conditions, the robot can reliably answer the question only 10% of the time, since no unique path exists from the maze arms.
Therefore, for assessing the quality of the perception space module, or map, or a model in general, we recommend generating localisation paths based on the second question. Since, the second question includes the notion of rewarding symmetry and aliasing with lower metric values, so rejecting apparently perfect solutions for potentially better ones, if they exist. It is also worth noting that a perception space scoring highly with the "Where I became lost?" question, is guaranteed to also score highly with the "Where Am I?" question too. However, the opposite is not true, as shown in the above example. Moreover, if the ideal set of criteria is satisfied, that of localising within one localisation step and within the desired localisation area, both of the localisation questions converge to the same solution, the solution where all of the Perception Areas are unique.

5. Experimental setup

The first three stages of the model are applied to the Pioneer robot that we have in the laboratory. The Pioneer is equipped with a set of wheel encoders, which it uses to provide odometry information, a set of ultrasound sound sensors, which it uses to freely wander around a target environment, Figure 6, and a panoramic vision sensor [4], which it uses to generate perception signatures.

The first stage of the model, as implemented on the robot, consists of an Omni-directional colour camera placed on the top and about the centre of the robot. A video sender sends the video information to a frame grabber connected to a standard PC, where the information is collected and processed. The frame grabber samples a 24-bit colour frame, at a resolution of 384 pixels by 284 pixels, at a frequency of 5Hz. The collected video data is very noisy and suffers from dropped frames, interference, "salt and pepper" noise and gaussian noise. Each raw frame is filtered with gaussian and median filters that remove some, but not all, of the 2-Dimensional noise. The video data is also greatly effected by lighting conditions. However, the conditions are acceptable for experimental purposes and were constant over the duration of the experiments.

Figure 6. A picture of a typical laboratory environment, this shows a well lighted, colourful environment covering an area of approximately 4 square meters.

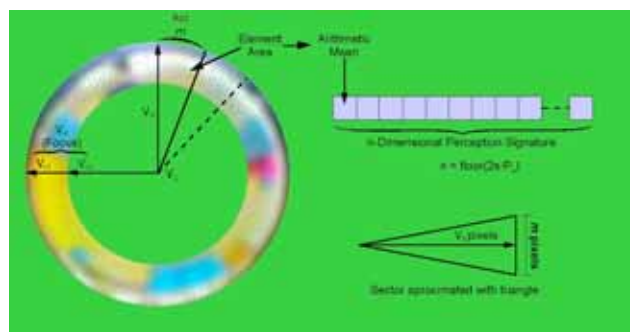

Figure 7. The raw panoramic image is divided into n sectors of arc m. Sectors are bounded by V_{r0} and V_{r1}, forming the Element Areas. The arithmetic mean is taken of each element area forming the perception signature elements. The background is a raw RGB perception signature bounded by $V_{r0} = 90$ and $V_{r1} = 130$, $V_R = 130$.

The filtered frames passed from the first stage are processed by the second stage into Perception Signatures. Frames are separated into six component frames of Red, Green, Blue, Hue, Saturation and Intensity and a perception signature generated from each. A Perception Signature is a 128-element vector of real numbers, the size of 128 is calculated to be the optimum. Segmenting a raw frame about its centre and taking the arithmetic mean of the pixel values within a bounded area of the segment forms the vector of real numbers, Figure 7. Each perception vector is cyclically shifted to minimise the Euclidean distance between a suitable reference vector, to account for rotational error.

Perception space modules process the perception signatures from the second stage. We have implemented four types of perception space module using standard classifier algorithms; a Leader classifier [7]; a K-Mean classifier [8]; a Growing Cell Structure [5] and a Fuzzy-C mean classifier [2]. The parameters associated with each classifier are fixed to suitable values, given in Table 1, but the maximum classes, N, a classifier can create is variable.

An instance of a perception space module is first trained with data from an initial exploration wander around the target environment, Figure 6. We collect a training data set over a 60-minute exploration wander, which produces six sets of perception signatures, containing approximately 16000 signatures each. An instance of a perception model is tested with data from subsequent explorations. We collect three sets of test data, from three separate explorations, each producing six sets of perception signatures, which contain approximately 5000 signatures each.

Table 1. Fixed parameters for the perception module classifiers

Classifier	Fixed Parameters
Leader	
K-Means	
Growing Cell	
Fuzzy C-Means	

Three perceptual area maps are generated from a trained perception space module, one for each test data set. A 10% random set of locations is chosen from within each perceptual map, and localisation paths calculated, $lp = 1500$. These localisation paths are used to evaluate the usefulness of the perception module using the *lost metric* benchmark described below.

6. Experimental results

Twenty-four implementation methods are created from the six perception signature types and the four perception module types. We describe how the *lost metric* is used to evaluate the usefulness of these twenty-four methods.

6.1. The *lost metric* benchmark

A *lost metric* value, based on a chosen set of *lost metric* criteria, is computed for each localisation path and the Root Mean Square Error is calculated, equation (4). This RMSE value is the perception modules benchmark value. A RMSE value of 0 represents the best case and a value of 1 the worst case.

$$f_{R.M.S.E.LostMetrics} = \sqrt{\frac{\sum_{i=0}^{lp-1}(1 - f_{i_{LostMetric}})^2}{lp}} \qquad (4)$$

6.2. The benchmark results

The aim of an implementation, or model in general, is to minimise the *lost metric* RMSE benchmark value. In this case, we wish to find the instance of the perception space module that does this. We have fixed all parameters associated with the four types of module, Table 1, and allow the classes to vary. A search is conducted within the class space to find the instance that minimises the RMSE benchmark value. The procedure applied to each implementation method is illustrated in Figure 8. The *lost metric* criteria values we used are $\omega = 5.0$, $\lambda = 0.5$, $t_w = 0.5$, $t_\lambda = 1.5$, $\theta = 0.9$, $t_\theta = 10$. Localisation area is expressed in robot areas units, t_λ is set to an area of 1.5 times the robot areas. The area of the Pioneer robot is 418cm^2.

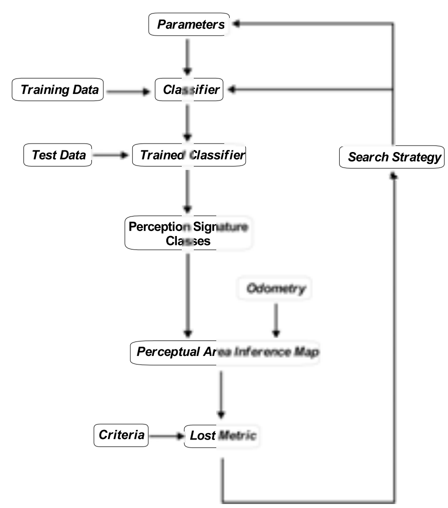

Figure 8. Searching the space of classifiers.

The results from the evaluation are recorded in Table 2. The table shows the instances of all the methods that have produced the minimum RMSE values, N is the number of classes in the perception module.

Table 2. Minimum RMSE values for the specified criteria

	Perception Signature Type					
Classifier	Red	Green	Blue	Hue	Saturation	Intensity
Leader	N: 78	N: 41	N: 267	N: 56	N: 358	N: 123
	M: 0.506	M: 0.505	M: 0.563	M: 0.490	M: 0.563	M: 0.514
K-Means	N: 35	N: 57	N: 43	N: 45	N: 83	N: 53
	M: 0.470	M: 0.463	M: 0.489	M: 0.470	M: 0.464	M: 0.462
Growing Cell	N: 85	N: 91	N: 85	N: 35	N: 59	N: 51
	M: 0.470	M: 0.513	M: 0.548	M: 0.473	M: 0.528	M: 0.483
Fuzzy C-Means	N: 274	N: 392	N: 76	N: 110	N: 280	N: 174
	M: 0.534	M: 0.622	M: 0.563	M: 0.512	M: 0.556	M: 0.548

Using the information presented in Table 2 we are able to select the best method, which is the module implemented with the K-Means classifier and the intensity perception signatures, under the given set criteria. The perceptual inference map from one of the test runs for this method is overlaid onto a stylised plan view of the target environment, Figure 9. The perceptual map in Figure 9 shows that the perception areas appear to be reasonable.

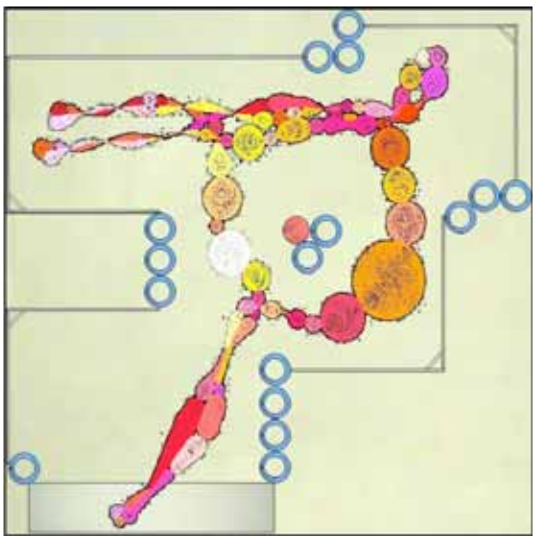

Figure 9. The Perceptual Area Inference Map generated by the K-Means module and the intensity signature, RMSE = 0.462, under the criteria of $\omega = 5.0$, $\lambda = 0.5$, $t_w = 0.5$, $t_\lambda = 1.5$, $\theta = 0.9$, $t_\theta = 10$. The inference map has been scaled and overlaid, matching the starting points, onto a stylised plan view of the target environment, Figure 6.

It can be seen that the perception space module with the smallest RMSE had a value of 0.462. In future work we will extend the search space to include all the classifier parameters in Table 1 and employ a genetic algorithm [6] search to attempt to reduce this value further.

The perception signatures describe above are sufficient for the purposes described here. However, they throw away much of the information in the original vision frame and are limited in the descriptive information they convey. We will further the research by investigating knowledge rich perception signatures [9], for example.

7. Conclusions

The biologically inspired model described in this paper was used to demonstrate how a new metric could be used to benchmark a range of implementations of the model. The *lost metric* is a useful tool that is used to express the usefulness of a model in terms pertinent to the navigation task, that of localisation. Moreover, where a model is defined by many parameters, the methods described here offer a method for automatically searching for the best solution. We feel this methodology and metric could be especially useful to robot constructors who need a method of optimising design parameters for particular applications We hope this metric will be adopted more widely, thus enabling quantitative comparisons to be drawn between the many spatial integration models that currently exist today.

8. References

[1] Bennett, A., "Do Animals Have Cognitive Maps", Journal of Experimental Biology, 199, 1996.

[2] Bezdek, J.C., "Pattern Recognition with Fuzzy Objective Function Algorithms" Plenum, New York, 1981.

[3] Cheng, K., "A Purely Geometric Module in the Rats Spatial Representation", Cognition, 23, 1986, pp. 149-178.

[4] Egerton, S.J., Callaghan, V., "A Biologically Inspired Mapping Model for Autonomous Mobile Robots", Advances in Intelligent Systems: Theory and Applications, Ed. Mohammadian, M., 2000, pp 209-218.

[5] Fritzke, B., "Growing Cell Structures – A Self Organising Network for Unsupervised and Supervised Learning", Technical report 93-026, International Computer Science Institute, Berkley, California, 1993.

[6] Goldberg, D.E., "Genetic Algorithms in Search, Optimisation and Machine Learning" Addison-Wesley, 1989

[7] Hartigan, J.A., "Clustering Algorithms", John Wiley & Sons, New York, 1975.

[8] MacQueen, J., "Some Methods of Classification and Analysis of Multivariate Observations", Proc. 5th Berkeley Symposium on Mathematical Statistics and Propability, 1967, pp. 281-297.

[9] Wahlster, W., "One Word Says More Than a Thousand Pictures. On the Automatic Verbalization of the Results of Image Sequence Analysis" Computers and A.I., 8, 1989 pp. 479-492.