Large Scale Environment Partitioning in Mobile Robotics Recognition Tasks

Boyan Bonev and Miguel Cazorla

Abstract—In this paper we present a scalable machine learning approach to mobile robots visual localization. The applicability of machine learning approaches is constrained by the complexity and size of the problem's domain. Thus, dividing the problem becomes necessary and two essential questions arise: which partition set is optimal for the problem and how to integrate the separate results into a single solution. The novelty of this work is the use of Information Theory for partitioning high-dimensional data. In the presented experiments the domain of the problem is a large sequence of omnidirectional images, each one of them providing a high number of features. A robot which follows the same trajectory has to answer which is the most similar image from the sequence. The sequence is divided so that each partition is suitable for building a simple classifier. The partitions are established on the basis of the information divergence peaks among the images. Measuring the divergence has usually been considered unfeasible in high-dimensional data spaces. We overcome this problem by estimating the Jensen-Rényi divergence with an entropy approximation based on entropic spanning graphs. Finally, the responses of the different classifiers provide a multimodal hypothesis for each incoming image. As the robot is moving, a particle filter is used for attaining the convergence to a unimodal hypothesis.

Index Terms—Visual localization, entropy, Jensen-Rényi divergence, classifier, particle filter.

I. INTRODUCTION

MOBILE robotics is a field with increasing number of applications and the range of possible environments for a robot is becoming wider. The capability of learning without the need of human aid is becoming a basic aspect in mobile robotics. Among the variety of machine learning applications to mobile robotics, an essential one is localization. The localization problem consists of estimating the position of the robot in a given map, based on the information provided by the sensors of the robot, such as cameras and range sensors, as well as the odometry of the robot, if available. This work presents a novel approach to image-similarity-based localization. The main purpose of the method is the scalability to large and complex environments, and this aspect is dealt with by dividing the domain in as many partitions as needed.

Cameras provide rich information and its correct interpretation is a complex and open problem. There are two well differentiated approaches to visual recognition: the structural-description models and the image-based models. The first

one is usually more complex and task-specific. The image-based recognition, also known as appearance-based recognition, relies on general features extracted from the image, ignoring the structures. In robotics this approach has already been used for localization. For instance, in [1] Menegatti et al. weight omnidirectional samples according to image similarity, to implement a Monte-Carlo localization method for constrained indoor environments. This technique is used for managing multi-modal probability density, for example when the current image matches more than one reference image. Still the complexity and size of the environment restrict the applicability of this method, because the image matching relies on a single similarity function of the Fourier coefficients of the reference images. Any classifier has a limited capacity which limits its scalability to deal with more complex patterns [2].

In this work we propose an unsupervised division of the sequence of reference images in several subsequences of images. For each one of the partitions we associate a similarity measure which yields an appropriate image retrieval result. We take as a similarity measure the euclidean distance in a suitable feature space which is automatically selected from a general set of low-level features. Two major questions arise from this approach: which division to perform on the sequence and, given a test image, which one of the several similarity measures to consider. For dealing with the first problem we make use of Information Theory. We estimate the local Jensen-Rényi divergence [3],[4] among the previous and the next images of the sequence and the divergence peaks determine the limits between two consecutive partitions. The Jensen-Rényi divergence is estimated in the feature space of the images, where the features are a set of low-level filters, similarly to a previous work [5], [6]. The second question we have to deal with, is how to put together the results generated by the different similarity measures. This problem is tackled with a particle filter [7], provided that the robot moves in some direction of the trajectory. In this experimental setup we are assuming that the robot will follow the same trajectory in which the reference images are taken. This assumption is inspired by a previous work [8] in which a robot performs vision-based navigation along corridor-like environments.

The rest of the paper is organized as follows. In Section II we explain the setup of the experiments. In Section III we present our information theoretic approach to data partitioning. Next, in Section IV we explain how we obtain a similarity measure for each different partition. Then in Section V we present a way put the results together and we show some results. In Section VI we conclude presenting our conclusions and future work.

Boyan Bonev and Miguel Cazorla are with the University of Alicante.
Dept. de Ciencia de la Computación e Inteligencia Artificial,
Universidad de Alicante
Apdo. 99, E-03080 Alicante, Spain
E-mail: boyan@dccia.ua.es

Fig. 1. Representation of the trajectory followed by the robot for taking the reference images. Sample omnidirectional images are shown. The route consists of a laboratory, a narrow corridor, a wide corridor, stairs, a hall, two short outdoor segments and a large one. The 3D-representation is courtesy of Juan Manuel Saez (University of Alicante).

II. EXPERIMENTAL SETUP

The experimental procedure for the presented system consists of making a camera-equipped mobile robot or a person follow some definite path or trajectory, in order to save video or a dense sequence of images. We call this sequence *reference images* because localization is with respect to them. In the presented experiments the size of sequence is 440 images taken along a 180m-long indoor/outdoor route which is shown in Fig. 1. In order to navigate, we use the method proposed in [8] because it is also based in omnidirectional images.

Once the images are collected, the system unsupervisedly performs a partition of the sequence. Then, a feature selection process is performed for each partition, in order to optimize each similarity function for its associated interval of images. The bank of filters used for selection consists of rotation invariant low-level filters such as edge detectors and color filters. Rotation invariance is possible because the camera we use is omnidirectional and is vertically oriented, as shown in Fig. 2.

The initial set of filters $\mathcal{F}$ consists of the responses of each image to the following filters:

- Nitzberg
- Canny
- Horizontal Gradient
- Vertical Gradient
- Gradient Magnitude
- 12 Color Filters H_i , $1 \leq i \leq 12$

The filters are applied to 6 different scales of the image. Moreover, the omnidirectional image is divided in 4 concentric rings and filters are also separately applied to each individual ring. This ring division and the whole feature extraction process are explained in more detail in [5] and are illustrated in Fig. 3. Therefore the filters bank $\mathcal{F}$ consists of $|\mathcal{F}| = 17 \times 6 \times 5 = 510$ features. Each image represents a point in this feature space which is invariant to rotation and is much less sensitive to small image variances than the raw

Fig. 2. The omnidirectional mirror is oriented to capture the ground and 360° of the surroundings. This orientation allows rotation invariance when rotating along the vertical axis. The omnidirectional mirror is the Remote Reality's OneShot360 and the video capturing device (mounted on the base of the lens) is a GreyPoint Flea2 firewire camera.

Fig. 3. The feature extraction process.

pixels space. The comparisons between images are performed in this feature space or a subset of it, as detailed in Section IV.

After the learning phase, we perform tests which consist of starting from some random position in the reference trajectory and moving in some direction, which can be forwards or backwards with respect to the direction of the reference trajectory. Localization along a trajectory makes sense in environments where the robot follows only forward/backward paths, like corridors, streets or avenues. Some navigation methods have already been developed for such situations. Concretely in [8] is described a previous work in which corridor-following navigation is performed using the same vision sensor that we use in this work. As the robot is navigating, it takes test

images at a fixed time interval. The first test image taken causes a multimodal response. The next images will make the system converge to a unique hypothesis for the position of the robot, with an error of ± 2 reference images. In the presented experiments about 10–15 images are necessary for achieving convergence. For larger sequences convergence to a unimodal hypothesis would be slower, keeping a low error.

III. DATA PARTITIONING

A. Motivation

Dividing the data in several partitions is a key to scalability. In this work we have to establish a similarity measure capable of indicating the most similar reference image to a new input image. This measure is formulated in terms of a distance between images in an appropriate feature space, further explained in Section IV. This formulation is equivalent to a *K-nearest neighbour* (K-NN) classifier where each reference pattern is a separate class. In the Machine Learning field it is well known that the capacity of a classifier is in a tradeoff with its generalization properties [9]. In other words, a large amount of data requires a high capacity, which incurs on an inaccurate classification of *test* patterns (those which do not belong to the training set or reference patterns). To avoid a classification accuracy decrease with the increase of the amount of training data we divide the data in as many partitions as needed. On the other hand a formulation with a single classifier is unable to handle the *perceptual aliasing* problem, which refers to different states producing similar sensor response. For example, a long trajectory could include corridors with identical appearance. In these circumstances the only way to know the correct location is to have into account the previous images, which is explained in detail in Section V. In the following Subsection we explain the criterion we propose for data partitioning.

B. Partitioning

Finding the optimal partitioning of the sequence of images would involve evaluating each possible subsequences configuration. As the optimal number of subsequences is also unknown, there are

$$\sum_{k=0}^N \binom{N}{k}$$

possible partition sets configurations for an amount of N reference images. In the presented experiments, $N = 440$ which means that there are $2,8392 \cdot 10^{132}$ combinations. The complexity of the problem makes it necessary to use some heuristics for finding a good partition set instead of the optimal one.

The heuristic we propose to use for this problem is based on Information Theory. The idea is to find discontinuities in the sequence of reference images in the feature space $\mathcal{F}$. The reason for this is that each partition has associated its own similarity measure, based on a subset of those features. For example, when the trajectory leaves the laboratory and the corridor begins, there is a significant discontinuity which we are interested in. Information Theory offers tools for

Fig. 4. Entropy divergence analysis of a single feature: the Nitzberg filter for the reference images 1–80. It can be observed that the filter response does not present any significant maxima or minima. However there is a change in the variability near to image #30. The entropy divergence measure lets us notice the change by presenting a high gradient at that point.

finding such interesting places in terms of entropy analysis. Entropy analysis allows us to measure changes in the amount of information in the data, regardless of data's nature. An example can be seen in Fig. 4. In this example a single feature is analyzed in order to show the idea of entropy divergence. However, we do not analyze single features but we estimate divergence in the whole feature space, as explained in the following subsection.

C. Jensen-Rényi divergence

In order to calculate the entropy divergence over the whole feature space of the images we need a way of estimating entropy in high-dimensional spaces. Traditionally this has been a drawback and entropy has usually been estimated in one or two dimensions because of the high computational complexity and estimation errors in spaces with more dimensions. However there are methods for entropy estimation which do not depend on the number of dimensions of the data. A widely used one is the estimation of Rényi entropy with *entropic spanning graphs*, described by Hero and Michel in [10]. In [6] we explain the estimation of Rényi entropy and further approximation to the Shannon entropy for calculating the Mutual Information with a feature selection purpose. Also, in [11] *entropic spanning graphs* are used for Mutual Information estimation for image registration. In this work we use the Jensen-Rényi divergence which is a more general criterion than Mutual Information, as shown in [3].

In [4] Hamza and Krim explain the properties of the divergence and show an example of edge detection for image segmentation. They show that it is nonnegative for $\alpha \in (0, 1)$ and it is symmetric and vanishes if and only if the probability distributions $p_1, p_2, \dots, p_n$ which it measures, are equal, for all $\alpha > 0$. The Jensen-Rényi divergence is symmetric and

Fig. 5. Sliding window for finding edges with the Jensen-Rényi divergence.

generalizable to any arbitrary number of probability distributions, which is not possible with other divergences, like the Kullback-Leibler one, for example. Also the Jensen-Rényi divergence gives the possibility to assign weights to the distributions.

Although Jensen-Rényi divergence is capable of measuring the divergence among several sets of data, in this work we use it for measuring the divergence between two distributions, for edge detection. We take a sliding-window approach, with two subwindows of equal size. The divergence of the two subwindows W_1 and W_2 is evaluated as the window slides, as represented in Fig. 5. The general formula of the divergence for n probability distributions $p_1, p_2, \dots, p_n$ is defined as:

$$JR_{\alpha}^{\vec{\omega}}(p_1, \dots, p_n) = H_{\alpha} \left(\sum_{i=1}^n \omega_i p_i \right) - \sum_{i=1}^n \omega_i H_{\alpha}(p_i), \quad (1)$$

where $H_{\alpha}(p)$ is the Rényi entropy, α is the Rényi entropy's parameter, and $\vec{\omega} = (\omega_1, \omega_2, \dots, \omega_n)$ is a weight vector which satisfies $\sum_{i=1}^n \omega_i = 1$ and $\omega_i \geq 0$ and α .

For the case of two subwindows, we have that W_1 slides gradually from region A to region B (Fig. 5). If p_a is the distribution of region A, p_b the one of region B, p_1 the distribution of W_1 and p_2 the distribution of W_2 , then we have that the parts of W_2 located in the regions A and B have partial histograms p_a and p_b with weights proportional to the sizes of the intersecting subregions. Let us say that $\lambda \in [0, 1]$ is the fraction of W_2 included in the region B, then $p_2 = (1 - \lambda)p_a + \lambda p_b$. This probability distribution is variable depending on the position of the window W . The subwindow W_1 lies entirely in the region A, so its distribution $p_1 = p_a$ and $p = (1 - \lambda/2)p_a + (\lambda/2)p_b$. Thus, the Jensen-Rényi divergence in function of λ is:

$$JR_{\alpha}(\lambda) = H_{\alpha} \left(\frac{1 - \lambda}{2} p_a + \frac{\lambda}{2} p_b \right) \quad (2)$$

$$- \frac{H_{\alpha}((1 - \lambda)p_a + \lambda p_b) + H_{\alpha}(p_a)}{2} \quad (3)$$

For $\lambda = 1/2$ the formula of the window's divergence is:

$$JR_{\alpha}(p_1, p_2) = H_{\alpha} \left(\frac{p_1 + p_2}{2} \right) - \frac{H_{\alpha}(p_1) + H_{\alpha}(p_2)}{2}, \quad (4)$$

The Rényi entropy (also known as α -entropy) of a probability density function $p(\vec{x})$ is defined as:

$$H_{\alpha}(p) = \frac{1}{1 - \alpha} \ln \int p^{\alpha}(\vec{x}) d\vec{x}, \quad \alpha \in [0, 1[\quad (5)$$

and it can be estimated by building the *minimal spanning tree* $MST(\{\vec{x}_i\})$ of the data $\vec{x}$ and computing the weighted length of its edges $\vec{e}$:

$$L_{\gamma}(\{\vec{x}_i\}) = \sum_{\vec{e}_{ij} \in MST(\{\vec{x}_i\})} |e_{ij}|^{\gamma}, \quad \gamma \in [0, D] \quad (6)$$

where d is the number of dimensions of the data. The following Rényi entropy estimator is asymptotically stable and consistent for $D \geq 2$, as showed in [12]:

$$H_{\alpha}(\{\vec{x}_i\}) = \frac{D}{\gamma} \left(\ln \frac{L_{\gamma}(\{\vec{x}_i\})}{N^{\alpha}} - \ln \beta_{L_{\gamma}, d} \right), \quad (7)$$

where $\gamma = D(1 - \alpha)$, N and D are the number of samples and the number of dimensions of the data $\vec{x}$, and $\beta_{L_{\gamma}, d}$ is a constant not depending on the probability function but on the graph minimization criterion. An approximation [13] that can be used for large d is $\beta_{L_{\gamma}, d} \approx \frac{\gamma}{2} \ln \frac{d}{2\pi e}$. This approach to Rényi entropy estimation has been successfully used in a previous work [6] where it is explained in more detail.

Finally, even though the α parameter is fundamental in α -entropy, it has a less significant effect on the entropy divergence (Eq. 4). If the misalignment between the distributions could be modeled accurately, then $\alpha = 0$ would correspond to the best choice as it generates a Dirac function at the matching point. In practice this is not a robust selection as a less peaked function is necessary for finding a maximum. On the other hand, $\alpha = 1$ is the most robust choice but it yields the function with least sharp peak. In the case of using the Jensen-Rényi divergence for segmentation purposes, several values of the α parameter have similar performance. In our experiments we set $\alpha = 0.8$.

To sum up, we can perform entropy divergence analysis like in Fig. 4 using a sliding window and calculating the divergence with Eq. 4. The approximation in Eq. 7 makes it possible to work with a large number of dimensions, which in our case is $d = |\mathcal{F}| = 510$ features. A question that arises now is: which size of the window is the most appropriate? It depends on the environment and on the distance between the images. However a multiscale analysis shows up that the discontinuities of interest remain with several window sizes, while those which do not interest us get displaced or disappear. See Fig. 6 where the divergence at 30 window sizes is represented. It is easily observed that at the images 241, 254 and 273, there are gradients in the divergence at all the scales while the other gradients get diagonally displaced. Therefore the criterion that we establish for dividing the sequence in partitions is the presence of strong peaks in the gradient function, for those gradients which are present at different scales. The result of partitioning the whole sequence of 440 images is shown in Fig. 7 where the multiscale divergence gradient is represented and the most peaked points are selected. There are 19 important discontinuities so 20 partitions are established. In Fig. 8 we show the physical position of the

Fig. 6. Top: A representation of the Jensen-Rényi divergence at 30 different scales for some images from the sequence. We can see that the X positions of some discontinuities get displaced at different scales (the scales refer to the window size) while some others persist. We are interested in the latter ones. Bottom: three sample images corresponding to the persistent discontinuities of the upper plot.

Fig. 7. The gradient of the divergence along the whole sequence of reference images. Some of the peaks are selected as significant discontinuities according to their sharpness.

discontinuities along the trajectory. In the next Section we explain how we perform localization for each single partition.

IV. LOCALIZATION IN EACH PARTITION

This Section explains how we obtain a similarity function which is adequate for a definite subsequence of reference images and how we use it for new images. Given a sequence of N images $\mathcal{I} = (I_1, \dots, I_N)$ which are associated to the linear positions along the trajectory $\mathcal{S} = (s_1, \dots, s_N)$, and given a test image I_T associated to a s_T position, the objective is to find an image similarity measure $M(I_i, I_T)$ which minimizes the error between the real position of the test image and the estimated position:

$$|s_i - s_T|, \forall s_T/s_1 \leq s_T \leq s_N \quad (8)$$

where $\hat{i}$ is the index of the reference image $I_{\hat{i}}$ which is the most similar (has the shortest distance) to I_T :

$$\hat{i} = \arg \min_i M(I_i, I_T) \quad (9)$$

Fig. 8. Left: The reference images which are selected as discontinuities are marked with a red circle. Blue boxes are an analogy with Fig. 1. Right: Some sample images. The third one corresponds to stairs, which cause several discontinuities near to each other.

Provided that the images are in a D -dimensional feature space $\mathcal{F} = \{F_1, \dots, F_D\}$ (already explained in Section II) we define the dissimilarity measure as the weighted euclidean distance in $\mathcal{F}$:

$$M_{\vec{\omega}}(I_i, I_j) = \sum_{d=1}^D (\omega_d [F_d(I_i) - F_d(I_j)])^2 \quad (10)$$

where the weights $\vec{\omega} = (\omega_1, \dots, \omega_D)$, $\omega_i \in \{0, 1\}$ determine which features are considered and which are not. These weights have to be set for minimizing the objective defined in Eq. 8. Intuitively, this minimization modifies the feature space so that the order of the images in the new space becomes more adequate in the sense that each image is similar to its actual neighbors and it is dissimilar to the rest of the images, as illustrated in Fig. 9.

In order to achieve a good generalization for new images, the maximum number of possible test images have to be considered. According to the definition of the problem (Section II) we have at our disposal only one sequence of N images for the training process. Therefore we have to separate it in train set and test set. The Leave One Out Cross Validation (LOOCV) evaluation method maximizes the number of tests. With this procedure the sequence is divided in a train set of $N-1$ images and a test of 1 image; the evaluation is repeated N times until every image in the sequence has played the test role. The following algorithm iteratively selects important features in a greedy order:

```

Input:  $\mathcal{I}, \mathcal{F}, N, D$ 
 $j = 0$ 
 $\omega_i = 0, \forall \omega_i \in \vec{\omega}$ 
while  $\exists \omega_i / \omega_i = 0$ 
     $\forall i / \omega_i = 0$ 
         $\vec{\omega}' = \vec{\omega}$ 
         $\omega'_i = 1$ 

```

Fig. 9. The feature space is modified so that each image lies near to their actual neighbours, and it lies farther from the rest of the images.

$$\varepsilon_i = \frac{1}{N} \sum_{T=1}^N |s_T - s_{\arg \min_k M_{\vec{\omega}'}(I_T, I_k)}|$$

$$\omega_{\arg \min_i \varepsilon_i} = 1$$

$$j = j + 1$$

$$E_j = \min \varepsilon_i$$

$$\Omega_j = \vec{\omega}$$

end

Return: $\Omega_{\arg \min_j E_j}$

The algorithm does not have a stopping criterion. Instead, it keeps on selecting features until all of them (D) are selected, and stores them in order, together with their associated classification errors. Finally, it returns the weights vector which has the lowest associated error.

Once performed the feature selection process, the dissimilarity measure for the set of images $\mathcal{I}$ is applied accordingly to Eq. 10 and the weights $\vec{\omega}$. When a new image arrives, it is said to be closest to some reference image $T_{\hat{i}}$, where $\hat{i}$ is the number of reference image which has the closest distance $M_{\vec{\omega}}$ to the new image, as expressed in Eq. 9. In Fig. 11 are represented the estimations of the 20 different similarity measures, for 220 test images which have not been used for the training process. It can be observed that for a single test image, each similarity function has a different response because each one is trained for a different subsequence of images. See Fig. 10 where a single classifier is trained for the whole sequence of images and for a small range of images (for a single partition). In the following Subsection we explain how we put these results together to obtain a single estimation.

V. LOCALIZATION IN THE WHOLE DOMAIN

In mobile robotics localization a very important source of information is the history of previous perceptions of the robot, as well as odometry, if available. When a new perception produces multiple hypothesis, as shown in Fig. 11, history can help to disambiguate and converge to a unimodal hypothesis. We tackle the problem with the classical Monte Carlo Localization (MCL), also known as *particle filter*. This is a Bayesian approach which aims to estimate recursively the

Fig. 10. Left: a classifier trained for the whole sequence of images. Right: a classifier trained for the images in the range 104–148 (a single partition of the whole sequence).

Fig. 11. Responses of 20 different similarity measures, each one trained for a particular subsequence of images. Note the coherence between test and reference images in the diagonal of the plot. Each test image produces 20 different hypotheses, however only one of them is coherent with the previous test images, if they are taken in a sequential order.

posterior distribution $p(s_{T_0}, s_{T_1}, \dots, s_{T_k} | I_{T_0}, I_{T_1}, \dots, I_{T_k})$, where $s_{T_0}, s_{T_1}, \dots, s_{T_k}$ are a sequence of hidden parameters and $I_{T_0}, I_{T_1}, \dots, I_{T_k}$ are the sequence of test images observed. Particularly we are interested in a marginal distribution of the posterior, which is called *filtering distribution* and is denoted as $p(s_{T_k} | I_{T_0}, I_{T_1}, \dots, I_{T_k})$. From this distribution we can obtain the posterior mode of the state, which in our case is the estimated position with respect to the reference images. This representation is approximate but it is nonparametric, which makes it possible to represent a wide range of distributions.

The MCL algorithm is described and analyzed in [7]. It samples the posterior distribution with a set of *particles* which correspond to positions in the trajectory. We take as many particles as similarity functions we have, however this number could be dynamically changed, according to the complexity of the distribution. For each newly obtained test image, the algorithm first resamples each particle according to a motion model. In our case the motion model is a bimodal distribution with mean ± 1 image and variance 2 images because we assume that the robot moves and takes test images at the same speed as in the training run, and we assume that it can move both forward and backward in the defined trajectory. We do not use odometry in the experiments we present.

Once changed the values of the particle positions, their weights have to be calculated, according to their likelihood. The likelihood is the probability of the perception, given the particle position. For example see Fig. 11 and let's say that a particle was sampled to the position 400 which is the last interval we have in the sequence, then the likelihood will be determined by the response of the similarity function denoted in the plot as $H1.$, which is $\arg \min_j M_{\vec{w}'}(I_{T_k}, I_j)$. It can be seen in the plot that if the perception I_{T_k} was actually coming from the image #200 of the test set, then $M_{\vec{w}'}(I_{200}, I_{400}) = 0$, so $j = 400$, which is the same value as the particle position and this would be the maximum likelihood possible. Otherwise, let us say that the perception came from the image #190, then according to the same similarity function, $j = 380$ and the likelihood is lower. Finally, if the perception came from some test image number lower than #170, the response would be rather random, as shown in the plot. In this case, even if we obtain a high likelihood for the particle in this iteration, in the following iterations it would get low because the new perceptions would be incoherent with the motion model.

Once calculated the weights of the particle, the algorithm performs an *importance sampling* of the particles. As many particles as needed (a constant number in our implementation) are sampled according to the likelihood of each one of the previously existing particles. This means that particles with a low likelihood will probably disappear, while locations with a high density of particles with a good likelihood will cause a higher concentration of particles. This can be observed in Fig. 12.

The fact that zones with a high likelihood attract the rest of the particles can mislead the algorithm to converge to some incorrect location. When this happens, after a few observations the likelihood of the particles gets very low. To get over such

Fig. 12. A Monte Carlo Localization trace of iterations 1, 5, 8 and 11. With the 11-th testing image the convergence to unimodality is achieved. The real position of the test images is represented with a red circle, while the particles are represented with blue asterisks. The likelihood of each particle is also represented.

situation a simple mechanism is introduced, which takes some particles with a low likelihood and places them randomly along the whole trajectory space. The number of randomized particles depends on the mean likelihood of all the particles. This way if all the particles have a high likelihood, little particles are randomized. This mechanism is also useful for the *kidnapped robot* problem, which consists of taking the robot from its location and make it jump to a different one.

In Fig. 13 is shown the performance of a single similarity function over the whole space of images, in contrast to the use of several functions, Fig. 11. It can be seen that the discontinuities present in a single similarity function are overcome with the partitioning approach.

VI. CONCLUSION AND FUTURE WORK

In this work we present a scalable visual localization approach which allows an autonomous robot to get localized over large trajectories. It is an approach based on machine learning and no assumptions are made about the training images. Only one training run over the trajectory is needed for the camera to collect all the reference images. We propose to use an information divergence measure for finding interesting places and partitioning the data. Then we propose to use simple specialized classifiers for each different partition. The off-line learning process for the presented experiment took about 10 minutes both for partitioning and for selecting the features for the different classifiers. Finally a unique localization estimation is obtained with a Monte Carlo sampling method. Classification of each new image can be performed online as its feature extraction process takes several cents of a second and the similarity measures evaluation is very fast. The results are promising, as shows the experiment with an indoor/outdoor trajectory of a 180m length, in which a unimodal hypothesis is usually achieved before the first 12 observations.

A future work is to extend this approach to 2D localization and to topological maps construction and localization. One of

Fig. 13. Top: The response of a single similarity function trained for the whole sequence of images. Bottom: Localization results at the test image #131, given by: a) Single similarity measure for the whole set of images; the response is marked with a red cross. b) Monte Carlo Localization response, the particles are represented with vertical lines. c) Groundtruth, represented with a red circle. It can be observed that for the single similarity measure, the localization result is far from the groundtruth, due to the discontinuity it presents at that point. However the MCL, which uses several similarity measures (Fig. 11), is closer to the groundtruth.

the contributions useful for topological navigation is finding interesting places in a quite data-independent way. These interesting places could be the nodes of a topological map, where a decision about which way to follow could be taken.

ACKNOWLEDGMENT

Thanks to Francisco Escolano and Juan Manuel Saez for their valuable help in the Information Theory field. This

research is funded by the project DPI2009-07144 from Ministerio de Ciencia e Innovacion of the Spanish Government.

REFERENCES

- [1] E. Menegatti, M. Zoccarato, E. Pagello, H. Ishiguro, *Image-based Monte-Carlo localisation with omnidirectional images*, Robotics and Autonomous Systems, Vol. 48, No. 1, pages 17–30, Elsevier, August 2004.
- [2] R.B. Fisher, *An Empirical Model for Saturation and Capacity in Classifier Spaces*, In proceedings of 18th International Conference on Pattern Recognition, Vol. 4, pages 189–193, August 2006.
- [3] A. Ben Hamza, H. Krim, *Jensen-Rényi divergence measure: theoretical and computational perspectives*, In proceedings of IEEE International Symposium on Information Theory, page 257, July 2003.
- [4] A. Ben Hamza, H. Krim, *Image Registration and Segmentation by Maximizing the Jensen-Rényi Divergence* EMMCVPR 2003 : energy minimization methods in computer vision and pattern recognition. Vol. 2683, pages. 147–163, 2003
- [5] B. Bonev, M. Cazorla, *Towards Autonomous Adaptation in Visual Tasks*, In proceedings of VII Workshop of Physical Agents, Las Palmas de Gran Canaria (Spain), pages 59–66, 2006.
- [6] B. Bonev, F. Escolano and M. Cazorla *Feature Selection, Mutual Information, and the Classification of High-Dimensional Patterns*, Pattern Analysis and Applications. Pp. 309-319. Vol. 11, No. 3-4. September 2008.
- [7] A. Doucet, S. Godsill, C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Statistics and Computing, Springer Netherlands, Vol. 10, No. 3, July 2000.
- [8] B. Bonev, M. Cazorla, F. Escolano, *Robot Navigation Behaviors based on Omnidirectional Vision and Information Theory*, Journal of Physical Agents, Vol. 1, No. 1, pages 27–35, September 2007.
- [9] I. Guyon, V. Vapnik, B. Boser, L. Bottou, S.A. Solla, *Capacity control in linear classifiers for pattern recognition* In proceedings of 11th International Conference on Pattern Recognition, volume II, pages 385–388, 1992
- [10] A.O. Hero, Bing Ma III, O.J.J. Michel, J. Gorman, *Applications of entropic spanning graphs*, IEEE Signal Process Magazine, Vol. 19, No. 5, pages 85–95, 2002
- [11] H. Neemuchwala, A. Hero, S. Zabuwala, P. Carson, *Image registration methods in high-dimensional space*, International Journal of Imaging Systems and Technology, Vol. 16, No. 5, pages 130–145, 2006
- [12] A.O. Hero, O. Michel: *Asymptotic theory of greedy approximations to minimal k-point random graphs*, IEEE Trans. on Infor. Theory, Vol.45, No.6, pages 1921-1939, 1999
- [13] D.J. Bertsimas, G. Van Ryzin: *An asymptotic determination of the minimum spanning tree and minimum matching constants in geometrical probability*, Operations Research Letters, Vol.9, No.1, pages 223-231, 1990