Accepted for publication in:
International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016

Image-based localization using Gaussian processes

Manuel Lopez-Antequera[1,2], Nicolai Petkov[1] and Javier Gonzalez-Jimenez[2]

***Abstract*—Visual localization is the process of finding the location of a camera from the appearance of the images it captures. In this work, we propose an observation model that allows the use of images for particle filter localization. To achieve this, we exploit the capabilities of Gaussian Processes to calculate the likelihood of the observation for any given pose, in contrast to methods which restrict the camera to a graph or a set of discrete poses. We evaluate this framework using different visual features as input and test its performance against laser-based localization in an indoor dataset, showing that our method requires smaller particle filter sizes while having better initialization performance.**

I. Introduction

Visual localization is the task of recovering the pose (position and orientation) of a camera from the appearance of the images that it observes, given a database of previously captured images and their poses. The topic is of great interest in robotics and hand-held applications in GPS-denied scenarios, as cameras are ubiquitous, cheap and no additional infrastructure is required. This problem is also known as visual place recognition, although this term is usually employed in the field of computer vision and refers to approaches that are limited to finding the most similar image from a collection of images, akin to content-based image retrieval. To illustrate the problem, let us consider the following example. Imagine the situation in Fig. 1, where images taken from positions *a* and *b* capture the ball and the monkey, respectively. If a new image is taken which captures the ball and the monkey, we can assume that the new image was taken at a nearby pose from *a* and *b*, (for example, at location *c*). Realizing this idea in software is not trivial, since images represented as a collection of pixels are not straightforward to compare: the visual appearance of a location can vary to a large extent depending on the exact viewpoint from which the image is taken, as well as other conditions such as illumination or changes in the environment.

Image descriptors transform an image (a grid of pixel values) into higher level representations or concepts. Traditional descriptors like SIFT [1] or SURF [2] describe local image patches using gradients and are deployed extensively in computer vision. State-of-the-art solutions in image description use Convolutional Neural Networks (CNN) to extract descriptors with a high level of abstraction (and robustness to visual appearance changes) to perform tasks such as image classification [3], [4]. These state-of-the-art descriptors can be used in tasks such as loop closure, but their use in localization is limited to nearest-neighbor approaches, unless traditional geometric features are calculated as well to perform registration.

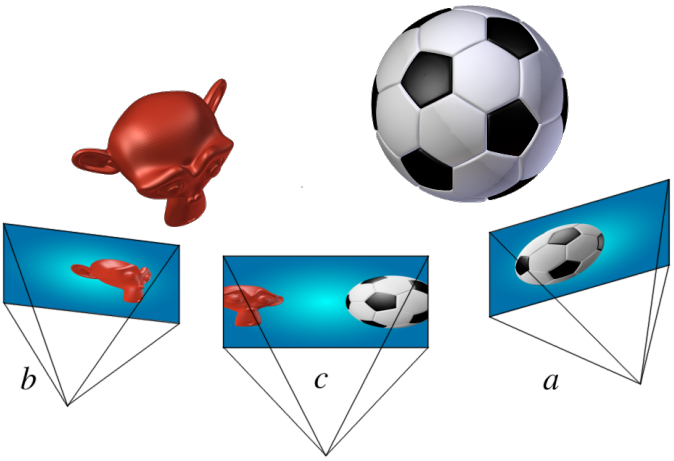


Fig. 1: Scenario in which cameras at locations *a*, *b* and *c* are capturing images. In this work we formalize the idea that images taken from nearby locations and orientations are expected to contain similar visual information.

In this work we attempt to overcome this limitation, performing localization using only whole-image descriptors. At the core of the method we propose the idea that images taken from similar poses should have similar visual content: when we look at a scene, the contents of the scene do not change drastically if we slightly rotate or move our heads: visual information enters or leaves the scene in a smooth, continuous manner.

For this reason, we model the appearance of images as continuous probabilistic distributions over all possible poses of the camera around the images in the database. In the previous toy example, this means that moving from location *a* to location *b* should yield a smooth change in the visual information (the ball slowly pans away from the frame as the monkey pans in).

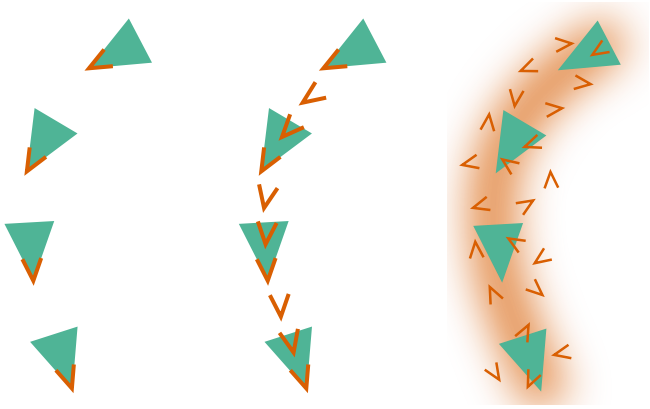
Through the use of Gaussian processes, our method is able to deliver a probabilistic estimate of what the visual information is at any unknown camera pose, provided that it is close enough to previously captured images. This results in a continuous localization system based on a sparse collection of keyframes of the environment. Unlike most visual place recognition systems which are restricted to a collection of previously recorded locations, our model provides continuous localization in all the pose space (see fig. 2), allowing

This work has been supported by the project "PROMOVE: Advances in mobile robotics for promoting independent life of elders", funded by the Spanish Government and the "European Regional Development Fund ERDF" under contract DPI2014-55826-R.

[1]Johann Bernoulli Institute of Mathematics and Computing Science, University of Groningen, The Netherlands n.petkov@rug.nl

[2]MAPIR-UMA group. Department of Systems Engineering and Automation, University of Malaga, Spain (mlopezantequera/javiergonzalez)@uma.es

Accepted for publication in:
International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016



(a) Discrete (b) Interpolation (c) All poses

Fig. 2: Most methods restrict the camera location. Our method allows the observation model to be evaluated at any position (orange) and orientation near the data (green).

for seamless combination with other continuous localization modalities such as laser or WiFi signal strength.

In this work, we introduce Gaussian processes for modelling visual information in a continuous manner over the space of 2D poses (fig. 2). This is then exploited for the task of image-based localization by using our framework as an observation model for a particle filter. In our experiments, we explore the viability of the method and we compare it against traditional laser-based particle filters, demonstrating faster convergence and greater robustness to the size of the particle filter.

II. Related work

If a large collection of images and their poses are registered into a consistent map using techniques like Structure from Motion [5], localization can be performed by simply querying the map for local feature matches and optionally checking for geometric consistency, such as in [6]. However, the limited descriptive power of local features means that as the map expands, individual features are not descriptive enough to find coherent matches, reducing their effectiveness in large databases, such as those extracted from large outdoor or multi-building scenarios.

For this reason, methods developed for CBIR (Content-based image retrieval) are usually preferred. These methods perform a global description of the image by generating 'holistic' or whole-image descriptors, more compact and uniquely descriptive than a collection of local descriptors. The disadvantage of using whole-image descriptors is the loss of detailed geometric information when local features are not stored, thus making geometric verification impossible. In these systems, the pose of the most similar image in the database (measured by descriptor similarity) is assigned as the current position of the camera.

Most of these global description models use visual bags-of-words to describe images by collecting histograms of local features such as SURF [2]. A direct approach is to find the most similar histogram in a database of previously collected locations, usually by first performing tf-idf weighting [7] so that features are weighted according to their relative frequency.

Visual localization is strongly related with the problem of Simultaneous Localization and Mapping (SLAM) in robotics. SLAM deals with the construction of a map of an environment while it is being explored, which consequently requires maintaining a correct localization of the robot. Many developments in visual localization come from this field. One of the most successful implementations of visual localization is used in the ORB-SLAM [8] system, which uses histograms of ORB [9] descriptors and then performs geometric verification only with the relevant features, instead of searching for individual descriptor matches in the whole database. A disadvantage of this localization model is that it is limited to the discrete locations from which the keyframes are taken.

A full 3D observation model is introduced in [10], by marginalizing out all of the observation likelihoods of individual landmarks (local visual features in 3D space) and then performing geometric verification. However, it relies on local features, suffering also from the aforementioned disadvantages.

A probabilistic approach is presented as part of FAB-MAP [11]. which builds upon the bag-of-words representation by defining a generative model. The method calculates the probability of being in each of the discrete locations of the map. It is a widely employed solution for the 'loop closure' problem (detecting if a robot is traversing a previously visited path).

All of the previously described methods treat locations and images interchangeably. This simplifies the treatment of the problem but limits localization to a discrete number of places/images. CAT-SLAM [12], [13] builds upon [11] by interpolating the probabilities along the edges connecting the positions of the database images in a graph. Through this approach, they allow the camera to be located in positions that are not part of the discrete set of images, although the positions are still restricted to the graph which connects the images' locations. (see Fig. 2)

To overcome this restriction, we employ Gaussian Processes to estimate the probability at any pose, not being limited to a graph or a discrete set of poses. Gaussian processes have been used for localization using WiFi signal strength as the sensing modality [14], [15]. In this work, we explore their use as a model for visual information.

III. Gaussian processes for modelling visual observations

We employ Gaussian processes to estimate the visual observation likelihood $p(\mathbf{z}|\mathbf{p})$ (i.e. *"how likely is the visual observation* $\mathbf{z}$*, given location* $\mathbf{p}$*"*), where $\mathbf{z} \in \mathbb{R}^k$ is the *observed* visual descriptor and $\mathbf{p} \in \mathrm{SE}(2)$ is the camera pose in 2D space.

In the following sections, we will first describe Gaussian Processes (GP) for a single variable, then discuss their extension to multivariate outputs and the use of locations and

Accepted for publication in:
International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016

orientations as inputs, all of which are necessary to correctly model visual observations.

A. Gaussian Processes

GPs[1] are non-parametric models which estimate the distribution of a function $\mathbf{p} = f(\mathbf{z})$ from a collection of training points $\{(\mathbf{p}_i, \mathbf{z}_i)|i = 1, ..., M\}$, and a certain measure of similarity given by the so-called kernel function.

A key element in GPs is that underlying knowledge about the model is not required. Instead, the correlation between points is specified through a kernel function $k(\mathbf{p}_i, \mathbf{p}_j)$, which only depends on the inputs $\mathbf{p}$ and a set of free hyperparameters. One of the most common kernels is the squared exponential or Gaussian kernel: $k(\mathbf{p}_i, \mathbf{p}_j) = \beta^2 \exp(-\alpha||\mathbf{p}_i - \mathbf{p}_j||_2^2)$. This is plotted in Fig. 3 for the unidimensional case, for different values of α and $\beta = 1$. This kernel specifies the correlation of any two points as being strong if they are near each other, exponentially decreasing as the norm increases. During training, the GP can estimate the values of these hyperparameters by finding the ones that best explain the data in the training set. When performing regression, the function f is estimated as a sum of all of the points in the training set, weighted by the kernel function.

Two important features of GPs are:

- Non-parametric model: no assumptions about the underlying model are made (as opposed to, for example, fitting the data to a linear model). Instead, a kernel function between pairs of points provides a measure of similarity.
- Treatment of uncertainty: the values of f obtained from the GP are accompanied by a measure of the uncertainty of the estimation, according to the data density around the query points and the kernel k (see Fig. 4).

We follow the notation from [16], where the training set of size M is defined as $\mathcal{D} = \{(\mathbf{p}_i, \mathbf{z}_i)|i = 1, ..., M\}$. In our application, the input vector to the GP is the pose of the camera in 2D: $\mathbf{p}_i = (x_i, y_i, \theta_i)$, and $\mathbf{z}_i$ is the visual descriptor vector of length k summarizing the image.

During the test phase, the GP performs regression on a test point $\mathbf{p}_i$. Since the uncertainty of the regression only depends on the kernel and the input data, it is the same for all the elements of the output [16]. Thus, the estimated value is represented by a k-dimensional isotropic Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}_i, \sigma^2 \mathbf{I}_k)$, where $\boldsymbol{\mu}_i$ is the vector of mean values and σ^2 is the variance.

B. Using poses as the input variables in a GP

In our application, the input variable $\mathbf{p}$ is not a single scalar, but a position and orientation on the map represented by a vector. As previously exposed, a GP can have any number of variables as input, as long as an adequate kernel function is provided. The kernel function must produce a measure of similarity between any two poses.

[1]Please, refer to [16] for a formal definition of Gaussian processes.

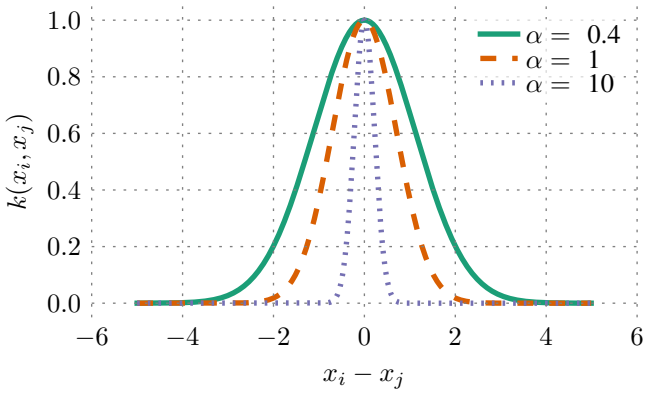


Fig. 3: Unidimensional Gaussian kernel, plotted for different values of α

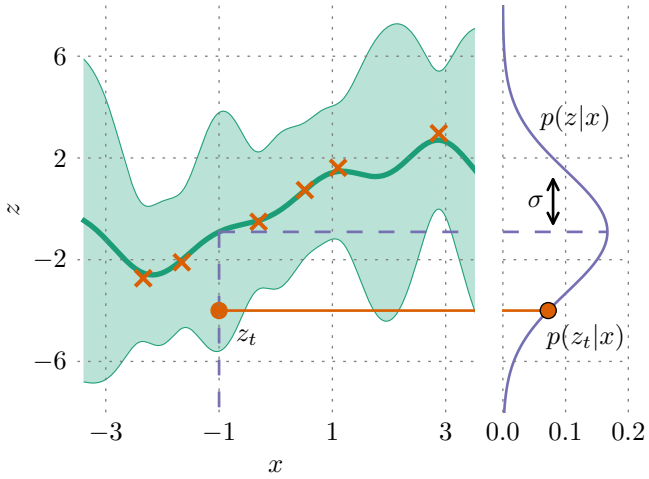


Fig. 4: A 1D Gaussian Process is trained on data (marked as '$\times$') and then used to perform dense regression on a range of x. The shaded region corresponds to a distance of 2σ from the mean. Note how the variance is smaller near the data. After training, we calculate the likelihood $p(\mathbf{z}_t|x)$ given an observation $\mathbf{z}_t$ and a location.

Let's denote the pose $\mathbf{p}_i = (\mathbf{x}_i, \theta_i)$. We select the following kernel to compare two poses $\mathbf{p}_i$ and $\mathbf{p}_j$ and produce a similarity measure:

$$k(\mathbf{p}_i, \mathbf{p}_j) = k_t(\mathbf{x}_i, \mathbf{x}_j) \cdot k_r(\theta_i, \theta_j) \tag{1}$$

where k_t is the Gaussian kernel (fig. 3), using the squared Euclidean distance between the two points (d^2) as the input:

$$k_t(\mathbf{x}_i, \mathbf{x}_j) = \beta_t^2 \exp(-\alpha_t \|\mathbf{x}_i - \mathbf{x}_j\|_2^2) \tag{2}$$

For the rotational kernel k_r, we also choose the Gaussian kernel, representing rotations as points in the circle S^1 through the mapping:

$$\mathbf{r}_i = \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \tag{3}$$

Accepted for publication in:
International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016

$$k_r(\theta_i, \theta_j) = \beta_r^2 \exp\left(-\alpha_r \|\mathbf{r}_i - \mathbf{r}_j\|_2^2\right) \quad (4)$$

Notice that this representation avoids problems caused by the ambiguity in angle representation. The product of kernels k_r and k_t leaves three hyperparameters to be estimated: α_r, α_t and the combined parameter $\beta = \beta_r \cdot \beta_t$.

IV. Observation model for particle filter localization

Particle filters (also known as Sequential Monte Carlo) are well-known in robotics for localization. At the core of particle filters, a collection of particles (likely states) represents a distributed hypothesis of where the robot is at any given time. These particles are randomly initialized and iteratively converge to the correct position through successive steps of weighting, resampling and motion.

1) Weighting: The robot senses the environment, and each particle is weighted according to the likelihood of that observation given the particle's pose.
2) Resampling: The particle set is resampled such that most likely particles are duplicated and least likely particles disappear.
3) Motion: When the robot moves, all of the hypotheses/particles move using the same motion. A noise term is added to each particle's motion to account for the uncertainty in its execution. This noise allows the newly duplicated particles to naturally separate from each other and create diversity.

When the location of the robot is unknown due to the system starting up or loss of tracking, the particle filter must perform a global initialization. On initialization, if no prior is available, all of the particles are drawn from uniform distributions, spanning all of the map area with random orientations. This process is usually the point of failure in particle filter localization systems, since enough particles must be used to cover all of the possible locations. A poor observation model can cause the filter to degenerate (to converge to a wrong location), causing catastrophic failure of the localization system. Instead, a well performing observation model will allow the particle filter to converge to the correct location in fewer iterations.

A. Observation model with GPs

One of the most common sensing modalities for particle filters is laser scans. During the weighting phase, the score of each particle is calculated according to the feasibility (likelihood) of the current laser scan, given the location of the particle in a 2D map of the environment. Instead, we propose using images (specifically, descriptor vectors extracted from images) to perform the weighting step.

As already shown in Section III, GPs estimate the likelihood of an observation given a trained model. Because of that, GPs fit seamlessly into the particle filter pipeline to perform the particle weighting. After acquiring an observation, the GP performs regression at each particle's location, obtaining as many predictions (and uncertainty estimates) as the number of particles in the filter. Particles are weighted according to the similarity between the estimated and the observed descriptors, considering the uncertainty of the estimation. In other words, particles will score maximally when the currently sensed descriptor is similar to the estimated distribution, if the estimate has high certainty.

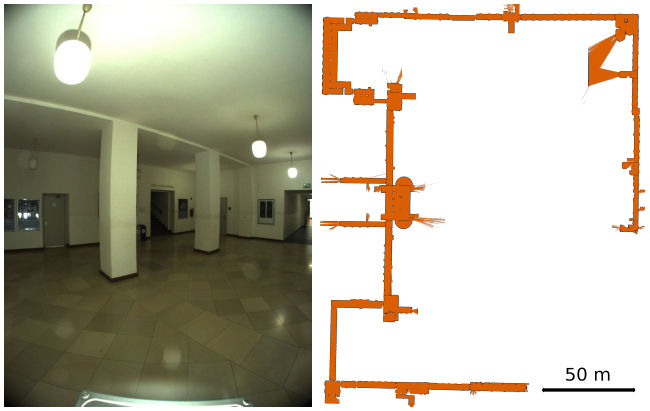


Fig. 5: Map of the dataset used for the experiments and a sample frame from the frontal camera

We train a GP using the kernel described in section III-B on descriptors extracted from a collection of images labeled with their positions, thus obtaining the parameters α_r, α_t and β. With these parameters the GP models the distribution $p(\mathbf{z}|\mathbf{p})$ of the visual descriptors in the pose space: For an arbitrary camera pose $\mathbf{p}_t$ the GP models the expected $\mathbf{z}$ as an isotropic Gaussian distribution $p(\mathbf{z}|\mathbf{p}_t) \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_k)$.

We illustrate this for the unidimensional case in Fig. 4. To weight the particles of the filter, we calculate the likelihood of the observation belonging to the distribution. The likelihood L is proportional to the probability of the occurrence of the observation $\mathbf{z}_t$ given the distribution $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_k)$ (i.e., z_i are assumed to be independent and identically distributed):

$$L \propto \frac{1}{\sqrt{(2\pi)^k |\sigma^2 I_k|}} e^{\left(-\frac{1}{2}(\mathbf{z}_t - \boldsymbol{\mu})^{\mathrm{T}} (\sigma^2 I_k)^{-1} (\mathbf{z}_t - \boldsymbol{\mu})\right)} \quad (5)$$

We calculate the log likelihood and drop the constant element $(2\pi)^k$ for convenience:

$$\ln(L) = -\frac{k}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(\mathbf{z}_t - \boldsymbol{\mu})^{\mathrm{T}}(\mathbf{z}_t - \boldsymbol{\mu}) \quad (6)$$

This expression can also be interpreted as a scaled, squared euclidean distance plus the $\frac{k}{2}\ln(\sigma^2)$ term:

$$\ln(L) = -\left(\frac{k}{2}\ln(\sigma^2) + \frac{1}{2\sigma^2}||\mathbf{z}_t - \boldsymbol{\mu}||_2^2\right) \quad (7)$$

V. Experiments

To evaluate the feasibility of our visual observation model as a weighting method in a particle filter, we perform a set of robot localization simulations using the TUMindoor [17] dataset. This dataset includes a 2D map of the environment (produced with a laser scanner) and images taken from a *Ladybug* omnidirectional camera rig, from which we only use the images captured from the front-facing camera. The images'

Accepted for publication in:
International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016

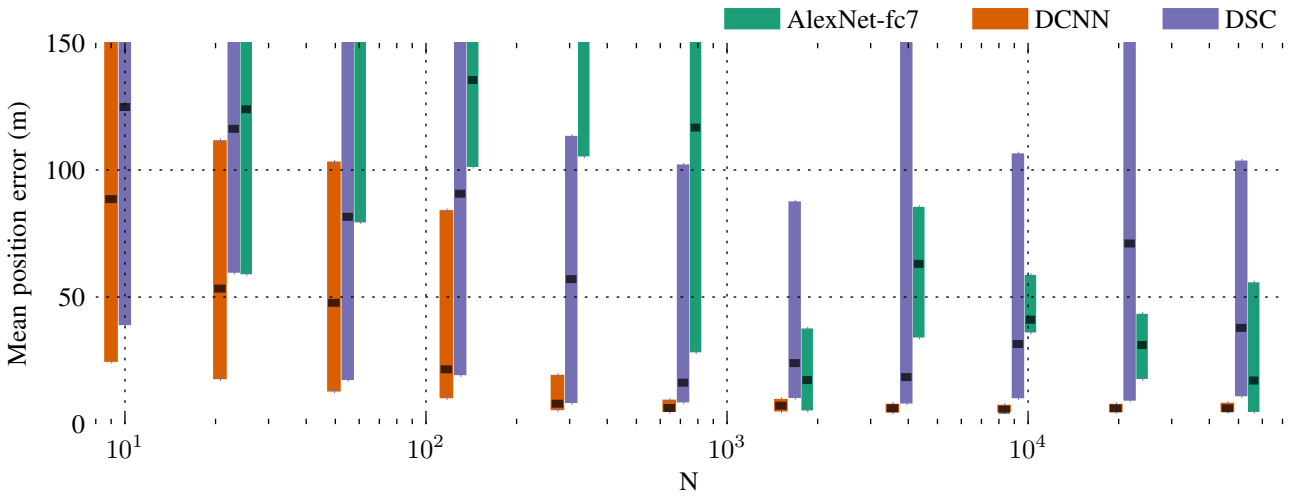


Fig. 6: Performance when comparing different descriptors as input to our method with respect to N, the size of the particle filter. The mean position error is taken after 100 iterations of simulation. The statistics for each box are calculated from 40 independent simulations. Each box represents the Q1-Q3 range and is marked by the median.

poses are also provided in the dataset. In particular, we train and test on different parts of the *2011-11-28* sequence, whose map and a sample frame are shown in Fig. 5.

We select a subset of the locations $\mathbf{p}_{train}$ and extract features $\mathbf{z}_{train}$ from the images at those locations. The GP's hyperparameters are found by fitting the model to this subset. The rest of the images and their descriptors $\mathbf{z}_{test}$ are then used to evaluate our approach for robot localization. The corresponding locations $\mathbf{p}_{test}$ are taken as ground truth. Within this real scenario, we perform simulations as follows. The camera starts at a random location from the test set $\mathbf{p}_{test}$. In each iteration, the next pose and descriptor are fed to the particle filter, skipping over the locations where the images for training were taken. Since the movement of the particles is performed exactly as in the ground truth, we add noise to the motion to allow the particles to diverge. This is performed by adding random noise to the rotation and translation of each particle.

Specifically, each particle's rotation angle is drawn from $\mathcal{N}(\Delta\theta_i, \sigma_r)$, where $\Delta\theta_i = \theta_i - \theta_{i-1}$ is the ground truth rotation increment of the sequence at time step i. Likewise, the particle's translation in $\mathbf{x}$ is drawn from $\mathcal{N}(\Delta\mathbf{x}_i, \sigma_t I_2)$. We set σ_r to 0.1 radians and σ_t to 0.5 meters in all of the experiments.

After each motion step, the particles are weighted according to the observation model being tested. The descriptor $\mathbf{z}_t$ from the observation at the current location $\mathbf{p}_t$ is compared with the GP regression $\mathbf{z}_i$ at each particle's location $\mathbf{p}_i$ as described in section IV-A. After weighting, normalization is performed by subtracting the minimum value and then performing division on the sum of the weights. Particles are then randomly resampled with a probability proportional to their normalized weight.

A. *Descriptor selection*

Until now we have not discussed the visual features used as input to our method since, in theory, it is agnostic to the type of holistic image descriptor being used. In practice, the features must reflect a characteristic explained in the introduction to this work: visual information does not change abruptly with smooth changes in the camera's location or orientation.

This is easy to interpret for humans, however, images represented as a collection of pixels don't follow this principle: a small change in position or orientation of the camera will make the value of each pixel change in an abrupt and nonlinear fashion, making our approach infeasible.

Several holistic descriptors have been used to perform CBIR (content-based image retrieval) and place recognition. The most successful ones at the moment are extracted from the intermediate representations of convolutional neural networks [18], [19].

A simpler and faster approach which has been proven to work in place recognition applications [20] is the use of a local-contrast-normalized and downscaled version of the image as a descriptor.

In our experiments, we test with several approaches based on convolutional neural networks, as well as the descriptor from [20] as a simple baseline. Specifically, we use:

- DSC: Downscaled and contrast normalized images as in [20].
- AlexNet: Generic CNN features extracted from the second-last fully connected layer (4096 elements) in the reference AlexNet network [4].
- AlexNet-PCA: A reduced descriptor of the AlexNet features (128 and 256 elements).
- DCNN: A short (128 elements) descriptor extracted from a convolutional neural network specifically trained for

Accepted for publication in:
International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016

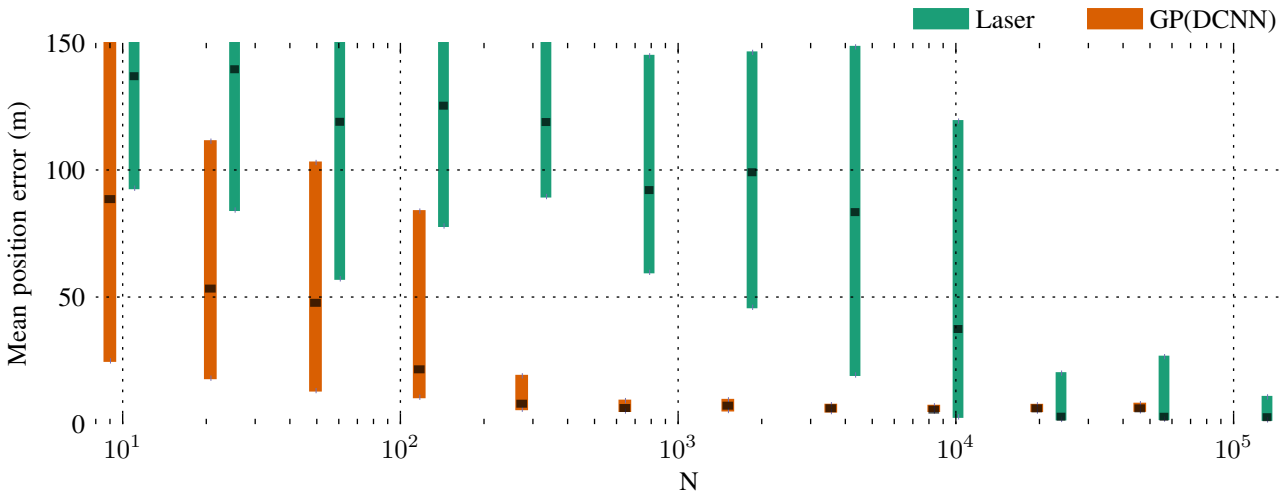


Fig. 7: Initialization performance when comparing with laser-based localization with respect to N, the size of the particle filter. The mean position error is taken after 100 iterations of simulation. The statistics for each box are calculated from 40 independent simulations. Each box represents the Q1-Q3 range and is marked by the median.

place recognition [19].

To select the most suitable image descriptor, we test them as input to our GP-based observation model in a localization simulation. In particular, we examine the performance when initializing the filter.

In all of the experiments, a random set of 30% of the frames is selected for training the GP. The rest of the images form the testing set, which is used to perform the particle filter simulation. In the case of this dataset, this means that the GP is trained with images which are, on average, separated 2.3 meters from each other. This is very sparse in comparison with traditional SLAM keyframes.

We test over 12 different settings for the particle filter size N, performing 40 simulations for each setting, for a total of 480 simulations per input descriptor. Each simulation begins at a random location from the dataset and continues for 100 iterations (100 consecutive locations in the test set).

From the tested descriptors, the best performing one was DCNN, which was specifically designed to compactly represent locations. The downscaled images (DSC) and the full AlexNet descriptor of 4096 elements achieved mixed results. Finally, the PCA-reduced versions of the AlexNet descriptor did not reach any significant results. We show the performance of DCNN, DSC and the full AlexNet descriptor in Figure 6.

B. Comparison with a laser-based observation model

After selecting DCNN as the most suitable descriptor for our method, we compare it against laser-based localization, which is widely used in indoor robotics. Since the TUMindoor dataset includes a 2D map of the environment, we can simulate laser scans at the test locations and perform weighting using the well-known likelihood field model [21].

Laser-based particle filters are usually stable once localized, but initialization can be troublesome, since laser scans aren't very descriptive (for example, laser scans from two different hallways might look quite similar).

We explore both methods and compare them attending to:

- Initialization / Relocalization
- Precision when correctly localized

1) Initialization performance: Increasing the number of particles N allows the filter to perform better, particularly during startup, when particle starvation can be problematic. However, the computational load increases linearly with the number of particles. The standard approach is to perform KLD sampling [22], which adaptively manages the size of the particle filter, reducing the number of particles when the filter is well localized.

In any case, with or without KLD sampling, when the filter is being initialized, a relatively large number of particles are required to successfully converge to the right location and to avoid particle deprivation, making relocalization costly in computational time.

A desired quality of an observation model is the reduction of this initial particle filter size. For this reason, we compare our observation model (using DCNN features) and the laser-based likelihood field model in simulation to ascertain their performance with respect to the size of the filter. Figure 7 shows how our observation model allows the particle filter to correctly initialize with a much smaller number of particles.

2) Localization precision: Both laser-based localization and our proposal have advantages and disadvantages: The results indicated in Figure 7 indicate how our method is better suited than the laser-based method for initializing a particle filter. However, when correctly localized, the laser-based method achieves greater precision. This can be seen in detail in Figure 8, where we only include simulations which are correctly localized (mean error under 10m after

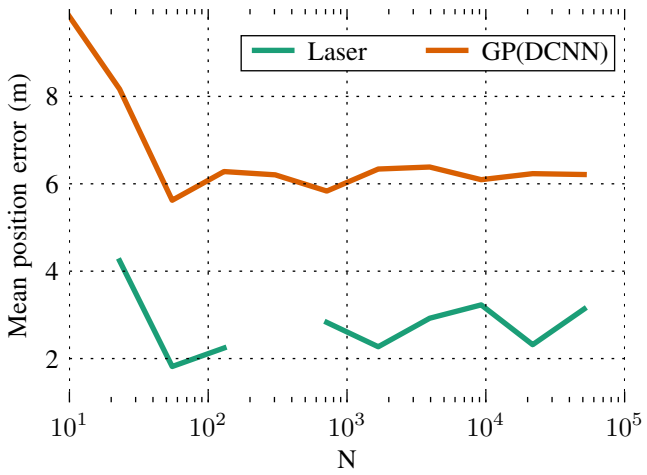


Fig. 8: Plot of the mean error of the simulations in which the particle filter converges (those that have achieved a mean error smaller than 10m after 100 iterations), both for our method and the laser based particle filter. We can see how laser based solutions are more precise (if correctly localized).

100 iterations). This opens up the possibility of combining both methods in future work.

VI. Conclusions and Future Work

Our work can be summarized as follows:

- We propose a probabilistic observation model for visual localization based on Gaussian Processes using appropriate kernels to model visual similarity in pose space.
- The model is not limited to the discrete locations where the images are taken, but is valid in all the possible positions and orientations around the data.
- We test different holistic descriptors as input. State of the art compact descriptors based on convolutional neural networks trained for place recognition tasks perform best with our method.
- Finally, we compare our proposal to a laser-based observation model, finding that our method can reliably localize the robot with fewer particles.

To our knowledge, this is the first proposal of an observation likelihood for images on the unconstrained continuous space of 2D poses.

This work could be expanded in several ways:

- Multi-camera systems or omnidirectional cameras could be used to increase the performance.
- Since the proposed observation model is probabilistic and continuous in the pose space, it is suitable for combination with laser or Wi-Fi signal strength modalities.
- The formulation could be extended to 3D movement, as long as suitable kernels are designed.

References

[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, nov 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *European Conference on Computer Vision (ECCV)*, Graz Austria, may 2006.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1106–1114.

[4] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features off-the-shelf: an Astounding Baseline for Recognition," *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pp. 512–519, mar 2014.

[5] J. L. Schönberger and J.-M. Frahm, "Structure-from-Motion Revisited," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] E. Deretey, M. T. Ahmed, J. A. Marshall, and M. Greenspan, "Visual indoor positioning with a single camera using PnP," in *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, oct 2015, pp. 1–9.

[7] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, oct 2015.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2564–2571, nov 2011.

[10] F. A. Moreno, J.-L. Blanco, and J. Gonzalez, "Stereo vision specific models for particle filter-based SLAM," *Robotics and Autonomous Systems*, vol. 57, no. 9, pp. 955–970, 2009.

[11] M. Cummins and P. Newman, "Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model," *International Conference on Machine Learning*, 2010.

[12] W. Maddern, M. Milford, and G. Wyeth, "Continuous appearance-based trajectory SLAM," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 3595–3600.

[13] ——, "Towards persistent indoor appearance-based localization, mapping and navigation using CAT-Graph," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, oct 2012, pp. 4224–4230.

[14] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Robotics science and systems*. Citeseer, 2006.

[15] M. Schussel and F. Pregizer, "Coverage gaps in fingerprinting based indoor positioning: The use of hybrid Gaussian Processes," in *Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2015, pp. 1–9.

[16] C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006.

[17] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach, "TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 1773–1776.

[18] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional Neural Network-based Place Recognition," *CoRR*, vol. abs/1411.1, 2014. [Online]. Available: http://arxiv.org/abs/1411.1509

[19] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. G. Jiménez, "Training a convolutional neural network for appearance-invariant place recognition," *CoRR*, vol. abs/1505.07428, 2015. [Online]. Available: http://arxiv.org/abs/1505.07428

[20] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1643–1649, 2012.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[22] D. Fox, "KLD-sampling: Adaptive particle filters," in *Advances in neural information processing systems*, 2001, pp. 713–720.