City-scale continuous visual localization

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

***Abstract*— Visual or image-based self-localization refers to the recovery of a camera's position and orientation in the world based on the images it records. In this paper, we deal with the problem of self-localization using a sequence of images. This application is of interest in settings where GPS-based systems are unavailable or imprecise, such as indoors or in dense cities.**

Unlike typical approaches, we do not restrict the problem to that of sequence-to-sequence or sequence-to-graph localization. Instead, the image sequences are localized in an image database consisting on images taken at known locations, but with no explicit ordering. We build upon the Gaussian Process Particle Filter framework, proposing two improvements that enable localization when using databases covering large areas: 1) an approximation to Gaussian Process regression is applied, allowing execution on large databases. 2) we introduce appearance-based particle sampling as a way to combat particle deprivation and bad initialization of the particle filter. Extensive experimental validation is performed using two new datasets which are made available as part of this publication.

I. Introduction

Performing self-localization with a single camera is of great interest in applications where GPS is unavailable or imprecise, as is the case in urban environments or indoor settings. Since it is a thriving research topic, many advances have been made recently [1], however, there are still limitations when dealing with:

- **Unconstrained topology of the database**: In order to develop systems that work online, the localization problem is usually posed as sequence-to-sequence or sequence-to-graph matching (especially in the case of appearance-based methods). Localizing efficiently in a database of unordered images is an open topic.
- **Changes in appearance** due to illumination or weather conditions. This leads to difficulties when comparing the input images to those from the database. This is particularly noticeable when using local feature descriptors such as SIFT.

To improve performance on these situations, we propose a method that leverages state-of-the-art convolutional neural network (CNN)-based descriptors to localize an image sequence taken from a monocular camera, using as reference an unordered, GPS-tagged collection of images (such as those readily available through Google Street View). Our proposal builds upon Gaussian process particle filters (GPPFs), in which Gaussian processes (GPs) are used as observation models for particle filters (PFs).

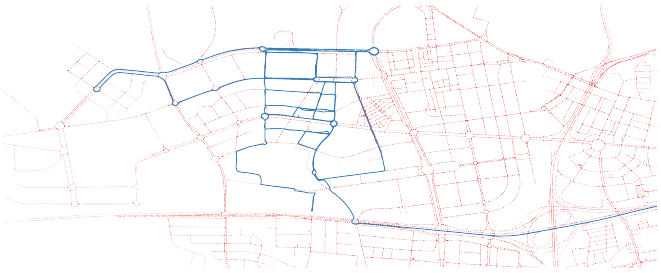

Fig. 1. Our contributions allow GPPFs to localize image sequences (blue, Málaga Urban Dataset [6] on large unordered georeferenced image databases (red, "Málaga Street View 2016" dataset, spanning 8 km^2).

GPPFs were introduced for signal strength-based robot localization in [2] and other modalities in [3], but their practical value for visual egocentric localization was limited at the time, as adequate image processing methods to exploit egocentric images within the framework were not available then. Now, recent advances from the computer vision community can be leveraged to enable egocentric localization through GPPFs. Specifically, we propose to use on whole-image descriptors extracted from convolutional neural networks trained for place recognition [4]. These representations are the state of the art in terms of robustness to illumination, weather, and long-term seasonal changes. An advantage of some of these features [5] is that they are trained so that their representations behave smoothly with respect to pose changes, that is, the distance between descriptors grows with increasing changes in camera pose. This behavior makes the descriptors amenable to interpolation over the pose space, which is desirable when used in a GPPF.

We expand upon previous work [7], in which GPs are used as an observation model for egocentric visual localization in an indoor scenario. Here, we introduce significant improvements to allow localization in large outdoor environments (8 km^2, Fig. 1) at interactive frame rates, while also enabling the system to handle global localization. Due to the small size of the image representations (8 kB per image), the system is scalable and feasible for portable applications. The main contributions of this paper are thus:

- The use of an approximation for GP regression (section III-A), enabling localization using GPPFs on large environments.
- The introduction of an appearance-based particle sampling scheme to enable the filter to initialize from an unknown location with a low number of particles

This work has been supported by the Spanish Government (contract DPI2014-55826-R), and the EU-H2020 project MOVECARE (Grant N. 732158).

[1]MAPIR-UMA group, University of Málaga, Instituto de Investigación Biomédica de Málaga (IBIMA), Spain (mlopezantequera/javiergonzalez)@uma.es

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

(section III-B).
- The collection of two new datasets: an unordered collection of 172.000 Google Street View images which serves as a map, and a collection of 50 sequences gathered from Mapillary[1].

We experimentally demonstrate our contributions in section IV by performing experiments which highlight the nature of these contributions and their effect on the success rate of global localization.

II. Related work

Pose representations

Space is continuous. However, for practical reasons, it is common to simplify appearance-based localization problems ("*where am I?*") by replacing them with classification problems ("*in which place am I?*"). Representing space as a discrete collection of places simplifies the problem: given a measure of image similarity, the most likely location is the one that is most similar to the current input. With this philosophy, FAB-MAP [8] is an approach to solve the place recognition problem by building a probabilistic model on top of a bag-of-words representation of images. Other methods exploit the sequentiality of the recorded images in the database and the live sequence, improving performance. In this line, SeqSLAM and its extensions [9], [10], [11] pose the problem as a sequence to sequence matching procedure, obtaining good results even with drastic appearance changes due to changing seasons. Similar work in [12] introduces efficient binary descriptors that allow direct sequence to sequence matching as a single hamming distance operation. The CAT-SLAM [13] system performs continuous localization: instead of discretizing the world into distinct places, they model the world as a continuous trajectory on which localization is performed. Although the probabilistic estimate of the position is a one-dimensional probability density function, however, localization is restricted to a sequence.

All of the previous methods constrain the problem to that of sequence-to-sequence localization, in which the database is formed by an ordered sequence of images. This restriction becomes problematic when dealing with scenarios where different trajectories are possible such as in a city, where many intersections exist and many routes cover the same locations. Some recent work deals with localization in such scenarios: In [14], the authors achieve localization of a moving camera in a city, however, they achieve this by representing the space as a dense grid, over which a Bayesian filter is applied. Although they achieve good results, representing the probability mass as a categorical distribution sets an upper bound of the size of the map. The authors of [15] achieve localization of a moving camera in a city by modelling the location of the vehicle as a categorical distribution on a graph of the road network. Using a graph representation of the city instead of a grid representation is advantageous, as memory and computation are not wasted on grid cells that represent non-transitable areas.

Image representations

Extracting representations that are useful for place recognition and visual localization is fundamental for any localization system. As many other applications within computer vision, visual localization has been improved dramatically by the use of CNNs, producing image representations that are robust to changes in illumination, weather and even the seasons: starting with [16], where the authors explored the use of internal representations of CNNs trained for object recognition. Later, [17] and [4] trained networks using semi-supervised, tripled-based training schemes to improve place recognition performance. Recently, the authors of [18] push the state of the art in place recognition by collecting a massive database of images from stationary webcams to train a CNN in a fully-supervised manner. Complementary to these advances, the work in [5] also applies CNNs to extract image representations that are tied to camera pose changes by linear transformations.

Gaussian processes for localization

GPs have also been used as an observation model to perform indoor Bayesian localization using WiFi signal strength [2], egocentric omnidirectional images [19] and egocentric monocular video [7]. More specifically, GPs within a PF-based localization (GPPFs) were introduced to the field of robot visual localization in [3], where the pose of a robotic blimp was tracked from an external viewpoint through a fixed camera. We build upon these works and extend the approach to large outdoor environments.

III. Gaussian Process Particle Filters

GPPFs are defined in [3] as PFs which use GPs for both the observation model and the transition model[2]. However, for self-localization of vehicles, it is not necessary to learn the transition model since wheel odometry is more reliable and is commonly available. Moreover, if the input frame rate is high enough, visual odometry (VO) can be used. The error incurred when estimating egomotion through VO is also well understood and does not need to be learned [20], [21].

GPs are a powerful tool to perform regression. It is out of the scope of this paper to introduce them[3], save for a short description: An intuitive view of GP regression is that predictions are calculated as a weighted average of neighboring points, where the weights are assigned according to a kernel function which provides a measure of distance or similarity of the query point to the neighboring training set points. GPs present two key features:

[1]Mapillary offers a crowdsourced collection of videos which are geotagged with poses refined using structure-from-motion techniques

[2]In a PF, an *observation model* predicts the observation for each particle. This prediction is compared to the real observation and determine the likelihood of a particle surviving. A *transition model* moves the particles according to some motion input. In some cases (for example, several degree of freedom actuators), the motion model can be learned from data, to help predict the actual motion from indirect sensing

[3]See [22] for a thorough reference on Gaussian processes

- GPs are non-parametric: instead of learning model parameters, the training data is used for regression.
- GPs output a probabilistic estimate of the uncertainty of the prediction.

As an observation model for a PF, the GP performs probabilistic regression, obtaining an estimate $\mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$ of the image descriptor $\mathbf{y} \in \mathbb{R}^D$ at any pose $\mathbf{p}_i = (x_i, y_i, \theta_i)$ in the plane. To this effect, a kernel function $k(\mathbf{p}_i, \mathbf{p}_j)$ must be defined to yield a measure of similarity. As in [7], we use the following kernel function to combine rotation and translation:

$$k(\mathbf{p}_i, \mathbf{p}_j) = \beta \exp\big(-\alpha_t \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 - \alpha_r \|\mathbf{r}_i - \mathbf{r}_j\|_2^2\big), \quad (1)$$

where $\mathbf{r}_i = (\cos(\theta_i), \sin(\theta_i))$, $\mathbf{x}_i = (x_i, y_i)$ and $\beta, \alpha_t, \alpha_r$ are the kernel parameters[4]. The observation model for the GPPF is the likelihood of the point belonging to the predicted Gaussian distribution. If all of the D dimensions of the descriptor $\mathbf{y}$ are assumed to be i.i.d, with standard deviation σ, we have:

$$p(\mathbf{y} = \mathbf{z}|\mathbf{x}) \propto \exp\big(-\frac{D}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}||\mathbf{z} - \boldsymbol{\mu}||_2^2\big) \quad (2)$$

In simple terms, particles whose predicted appearance is similar to the observation score high as long as there is confidence about the predicted appearance. For this observation model to work properly, the chosen image descriptor must be amenable to interpolation, that is, the values of the elements of the descriptor should behave smoothly with small camera pose changes. Descriptors extracted with CNNs trained to perform place recognition are well suited for this [7].

To perform localization, the GPPF iteratively carries out the following steps. 1: Particles are moved, following some motion input (e.g. wheel odometry). 2: Particles are scored with the observation model (eq. 2). 3: Particles are resampled: Those with higher score have bigger chances of being sampled. We now introduce two improvements to this system to enable online global localization in large environments.

A. Fast GP regression

GP regression becomes intractable when the size of the database n increases, due to their quadratic and cubic increase on compute time and memory use, respectively. In the context of outdoor visual localization in a city where the state can be any pose (x, y, θ), we can expect that a certain density of data points will be required to achieve localization. The value of this density will define an upper bound on the size of the world that the system can work in. Several approaches to reduce the time and memory requirements of GPs are discussed in [22], most of which reduce the complexity by replacing the training set with a different, smaller set of points $m < n$ that is used for inference. We choose the simplest of these, called Subset of Datapoints approximation in [22]. In this approximation, only a subset of the datapoints is used to perform inference.

[4]Although the GP kernel parameters and noise variance can be learned from data, we have empirically picked the following values for all of the experiments: $\alpha_t = 12$, $\alpha_r = 0.025$, $\beta = 0.5$, $\sigma_n^2 = 4$

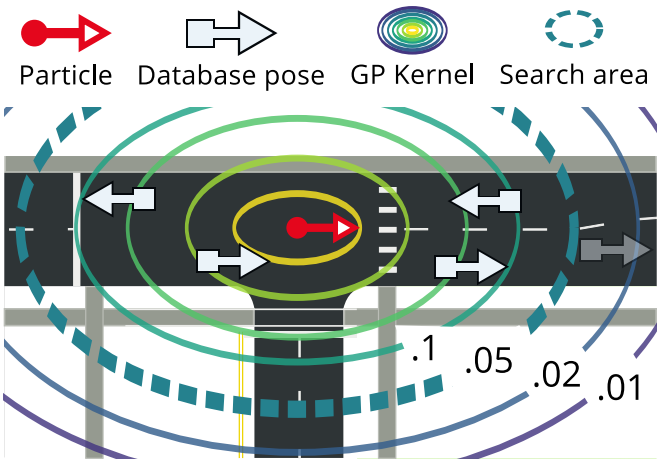

Fig. 2. Approximated GP regression allows the filter to work in large environments. The approximation only uses points that are close (in x, y, θ) to the particle being weighted. The value of the GP kernel is used to define a region from which to select these points. In this illustration, simplified to two dimensions x, y, only points in the area with kernel values under .05 are included. The shaded database point, as well as any other points in the database not seen in the figure, are not used to weight this particle.

In the general case, this approximation can be difficult to implement correctly: the criterion for selecting which subset of points to use is not always simple. However, for this application and the selected Gaussian kernel, selecting which datapoints to use can be done effectively and efficiently, since that only points that are located close enough to a given particle will have an effect in the regression of the descriptor at that particle's location. This can be seen intuitively: images that are far away in position or orientation (for example, rotated more than 90 degrees or 1 km away) have nothing to contribute to the output. We implement this by indexing the locations of the images of the database in a k-d tree. During the execution of the PF, the neighboring datapoints for each particle are searched (Fig. 2) and used as part of the GP observation model, while the rest of the database is ignored. Since the datapoints from the reference database are evenly spread over the map, the weighting phase of the PF executes in constant time regardless of the area of operation. The time of the search does depend on the size of the map, but it is small and grows, at worst, linearly with the number of datapoints in the map [23].

B. Appearance-based particle sampling

When the filter is initialized with an unknown position of the camera, particles are scattered over the map. After that, at least one particle must be close to the right location for the filter to be able to converge. If the map is large, this means that a large number of particles must be used so that the space x, y, θ is densely covered.

Adapting the number of particles so that they are reduced when the filter converges has been a successful solution for indoor, laser-based localization systems [24]. However, on a large outdoor environment like a city, the amount of memory and computation time required to cover the pose

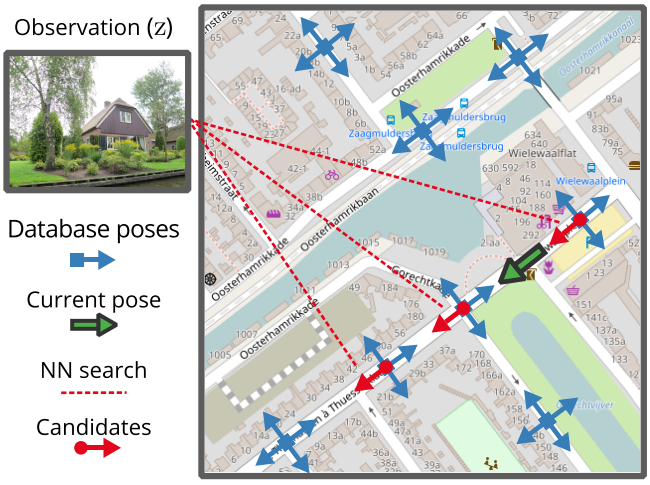

Fig. 3. Drawing new particles from appearance-based nearest neighbor proposals allows the filter to perform global localization and to escape wrong convergence.

space sufficiently makes this unfeasible. Another common problem with PFs is that they can converge to a wrong solution, leaving the filter in an unrecoverable state.

Traditionally, these issues have been relieved by introducing particles at random locations at every evaluation of the PF. We also propose to sample particles at new locations not previously represented by the probability mass. However, instead of sampling randomly, we generate candidates at locations which are visually similar to the current observation (see fig. 3), exploiting the fact that descriptors extracted from CNNs are suitable for appearance-based image retrieval [25].

During the resampling phase of the particle filter, images similar to the current observation in the database are searched: The n_a nearest neighbors of the descriptor z of the current image are retrieved. Then, with probability p_a, particles' poses are set to one of these nearest neighbors (chosen randomly), instead of being resampled from the existing probability mass. This method allows the filter to perform global localization and to recover from incorrect convergence. Another advantage is that the system does not need to explicitly detect that it is lost: the same operations are performed at every PF iteration. This search is also accelerated by means of a k-d tree, so that its time complexity is, at worst, linear with the size of the image database.

IV. Experimental evaluation

In this section, we first introduce the datasets used to perform our experiments: two new datasets and an already existing one. We then perform experiments analyzing the effects of fast GP regression and appearance-based sampling. Finally, we test our system on a challenging crowdsourced collection of sequences.

Datasets and image representation

All our experiments are performed with datasets from the city of Málaga (Spain). We have gathered two new datasets and also use an existing sequence.

Málaga Street View 2016: In order to have a database of images covering a large surface in which to localize video sequences, we collected images in an area of 8 km^2 surrounding the main campus of the university of Málaga using Google Street View. Four images were collected at each location where a Street View panorama was available: facing the vehicle's orientation, and at 90, 180 and 270 degrees. The database, shown as red points in figure 1, is composed of 172.000 images from 43.000 locations.

Málaga Mapillary 2017: We downloaded 50 sequences of images from Mapillary, selected so that they overlap with the Málaga Street View 2016 dataset (used as reference). We selected sequences whose ground truth poses met either one of these criteria: a) Sequences of 20 or more frames in which at least 80% of the images are within the bounding box of the reference database. b) Sequences where 100 or more frames are within the bounding box of the reference database, regardless of the total length. We discarded sequences with wrong or no compass information[5]. This dataset is intended to be used as a difficult test case for localization, as the sequences are recorded in uncontrolled conditions: different cameras, modes of transport, times of day, points of view, speeds, etc.

Málaga Urban Dataset (2013): We also rely on the Málaga Urban Dataset [6] as an easier sequence on which to localize (when compared to the Mapillary sequences), as it is long and recorded from a forward-facing viewpoint on a stable platform. It is sourced from video recorded with a Bumblebee 2 stereo camera mounted on a car. The sequence was recorded on a single 37 km run and includes precise ground truth location from RTK GPS.

Image representation: On all our experiments, we extract NetVLAD [4] descriptors to represent images, following preliminary results where "off-the-shelf" CNN representations and other compact descriptors for place recognition [17] did not work as reliably. The dimensionality of the NetVLAD descriptors is reduced from 1024 to 128 elements through principal component analysis (PCA). This reduction is computed on the reference database (Málaga Streetview 2016) and applied online to the images of the test sequence.

Experiment 1: Fast GP regression

To evaluate the effect of the subset of data approximation, we select random entries (image descriptors and poses) from the Málaga Street View 2016 dataset. We then predict their values through GP regression, using a variable number of neighboring points as data. We compare the result of performing GP regression using a small number of points $\mathbf{y}_{fastGP}$ with the result obtained using a large number of points $\mathbf{y}_{GP}$ (since using the whole dataset is not possible on a normal desktop computer due to memory constraints, we select a 'large' number of points by picking all points within 100 m of the query). We record the normalized euclidean distance from the result of the approximated GP regression

[5]We assumed wrong orientation if it differed by more than 30 degrees, on average, from the orientation of the vectors pointing from the location of each point to the next one in the sequence

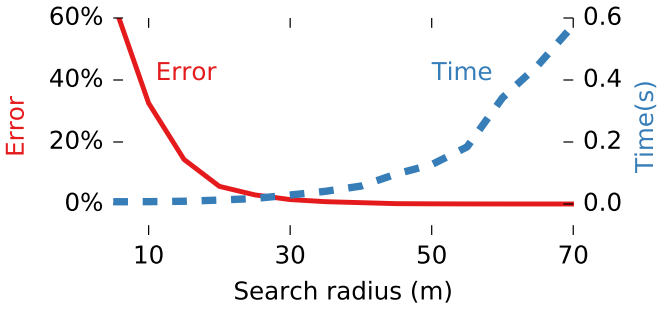

Fig. 4. Using only the neighboring points for GP regression is sufficient on the Málaga Street View 2016 dataset and enables timely execution.

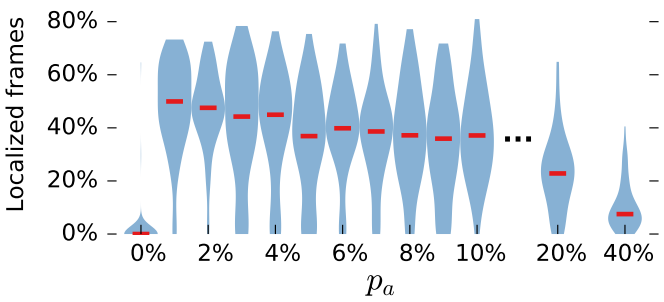

Fig. 5. Sampling a few particles from the reference database at each iteration based on their appearance enables global localization. If too many particles are sampled this way, the filter degenerates into frame-by-frame appearance-based place recognition

to that of the 'full' GP, $||\mathbf{y}_{fastGP} - \mathbf{y}_{GP}||/||\mathbf{y}_{GP}||$ for each test case. Results are averaged over 100 test samples and shown in figure 4. As expected, error decreases when the search radius is increased, also increasing the computational demand. More importantly, selecting a radius larger than 30 m yields almost no error reduction, validating the use of this approximation for localization. We fix the search radius to this value in the following experiments.

Experiment 2: Appearance-based particle sampling

We now test the added value of appearance-based sampling of new particles as introduced in section III-B. We do this by evaluating the full localization system, using the Málaga Street View 2016 dataset as reference, and the Málaga Urban Dataset as the test sequence (both shown in figure 1). The problem is reduced to 2D localization by projecting the poses of the database and the test sequences onto a 2D plane tangential to Earth's surface at the mean point of the locations in the reference database. The PF is initialized by uniformly scattering particles on the map. The size of the filter is set to 500 particles in all our experiments. To simulate errors in motion sensing, the ground truth motion between consecutive frames in the test sequence is perturbed by noise[6] before being used as the odometry input. Particles are moved with the same motion model doubling the amount of position and rotation noise that is added to the actual input. This is done in order to enforce diversity in the particles' poses. The particle filter is evaluated (weighting and resampling) after every 5 m of motion according to this simulated odometry. The output of the system is calculated as the mode of the distribution, estimated by running mean shift on the position of the particles with a Gaussian kernel of $\sigma = 20\ m$. The system is considered to have localized correctly if this estimate is within 15 m of the ground truth position. In each run of the simulation, a randomly selected section of the Málaga Urban Dataset sequence is used, effectively testing on different subsets of the test sequence. Each simulation is executed over 1000 consecutive frames.

We test the effect of appearance-based sampling by varying the values of the parameters p_a and n_a and observing their effect on the localization performance. In fig. 5, we plot the fraction of localized frames in the sequence over 100 particle filter simulations for each value of p_a. The figure shows how completely disabling appearance-based sampling (p_a = 0) makes it very difficult for the PF to localize, as it is highly unlikely that a particle is randomly sampled at the correct pose during initialization. Enabling appearance-based sampling by selecting a small value of p_a allows the newly sampled particles to drive the distribution close to the ground truth location, however, if p_a is large, then many particles are sampled based on image appearance on every step, making the distribution of particles frequently 'jump' from location to location, discarding any accumulated evidence. The effect of the value for n_a is not shown in the figure, since we found the method to be quite robust to the specific value of the number of neighbors within a range $2 < n_a < 10$.

Experiment 3: Localization of crowdsourced sequences

We evaluate the localization system using both improvements (fast GP regression and appearance-based particle sampling) by performing localization of the sequences from the Málaga Mapillary 2017 dataset. This experiment has the same structure as experiment 2, fixing $p_a = 1\%$ and $n_a = 2$. These sequences are more challenging than the Málaga Urban Dataset [6], since they were captured in unconstrained conditions and vary in length from 100 m to 5.6 km, the shorter ones being more difficult to localize as the filter has less chances to accumulate evidence.

We test our system on these sequences and compare against a baseline where each particle is directly weighted using the descriptor distance to the closest image in the database, that is: $w = e^{-||\mathbf{z}-\mathbf{y}_{\mathrm{NN}}||_2}$, where $\mathbf{y}_{\mathrm{NN}}$ is the descriptor of the image in the database closest to the particle being weighted[7]. This baseline uses the same image representation as our proposal (PCA-reduced NetVLAD). We also endow it with appearance-based particle sampling (Sec. III-B). Otherwise, global localization is nearly impossible on this dataset. This comparison thus highlights the advantage of performing probabilistic regression instead of a simple

[6]Gaussian noise with $\sigma_d = 0.1d$ is added to both elements x, y of the motion vector, where $d = ||(x, y)||_2$. The orientation of the particles is also perturbed by Gaussian noise with $\sigma_r = 0.05|r|$, where r is the angle of rotation of the ground truth motion.

[7]we first search for the four closest images and then pick the one with the most similar orientation

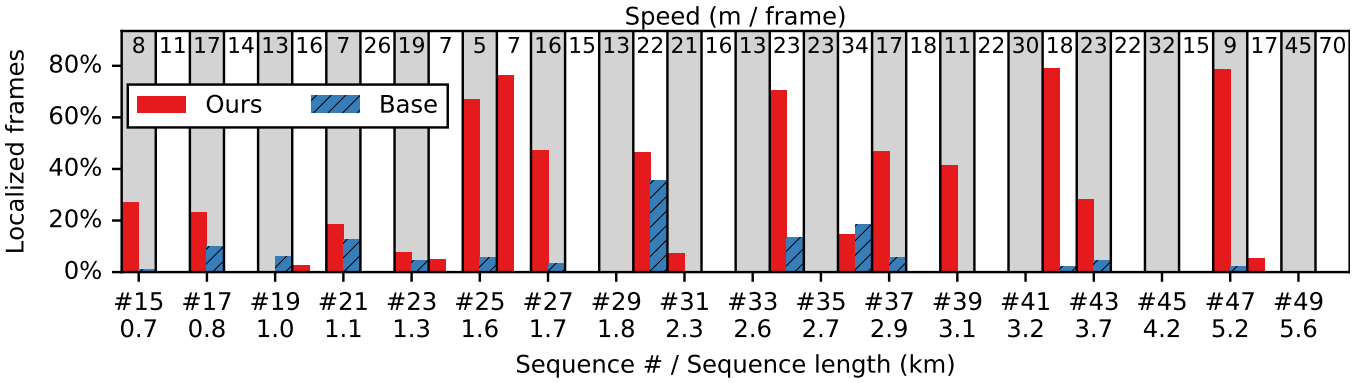

Fig. 6. Fraction of localized frames in sequences 15 to 50 of the Málaga Mapillary 2017 dataset, averaged over 20 runs.

image-to-image comparison when performing localization, as all other aspects (particle filter, motion model, image description, resampling scheme...) are the same. Results are shown in figure 6 as the average number of localized frames for 20 runs on sequences 15 to 50. Sequences 1 to 14 are shorter (under 700 m) and neither the baseline nor our method achieved localization.

V. Conclusions

In large environments, global localization with a standard particle filter is infeasible using a normal GPPF. The appearance-based sampling introduced in section III-B enables global localization with a small number of particles by exploiting appearance-based retrieval techniques. The use of a subset of data approximation allows evaluating the observation model in linear time instead of quadratic time, making GPPFs feasible in large environments.

Experimental validation shows that these advances enable the use of GPPFs for practical, online localization based on egocentric images. As part of this publication, we offer the Málaga Street View 2016 and Málaga Mapillary 2017 datasets online at `mapir.isa.uma.es`.

References

[1] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics (TRO)*, 2016.

[2] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *Proceeding of Robotics: Science and Systems*, 2006.

[3] J. Ko and D. Fox, "GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[4] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] D. Jayaraman and K. Grauman, "Learning image representations tied to egomotion," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[6] J.-L. Blanco, F.-A. Moreno, and J. González-Jiménez, "The Málaga Urban Dataset: High-rate Stereo and Lidars in a realistic urban scenario," *The International Journal of Robotics Research (IJRR)*, 2014.

[7] M. Lopez-Antequera, N. Petkov, and J. Gonzalez-Jimenez, "Image-based localization using Gaussian processes," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016.

[8] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research (IJRR)*, 2008.

[9] 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)*, 2012.

[10] E. Pepperell, P. Corke, and M. Milford, "All-environment visual place recognition with SMART," *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[11] ——, "Routed roads: Probabilistic vision-based place recognition for changing conditions, split streets and varied viewpoints," *The International Journal of Robotics Research (IJRR)*, 2016.

[12] R. Arroyo, P. F. Alcantarilla, L. M. Bergasa, and E. Romera, "Towards Life-Long Visual Localization using an Efficient Matching of Binary Sequences from Images," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[13] W. Maddern, M. Milford, and G. Wyeth, "CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory," *The International Journal of Robotics Research (IJRR)*, 2012.

[14] G. Vaca-Castano, A. R. Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] A. Taneja, L. Ballan, and M. Pollefeys, "Never Get Lost Again: Vision Based Navigation Using StreetView Images," in *Asian Conference on Computer Vision (ACCV)*, 2015.

[16] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional Neural Network-based Place Recognition," *arXiv*, 2014.

[17] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. Gonzalez-Jimenez, "Training a Convolutional Neural Network for Appearance-Invariant Place Recognition," *arXiv*, 2015.

[18] Z. Chen, A. Jacobson, N. Sunderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep Learning Features at Scale for Visual Place Recognition," *arXiv*, 2017.

[19] T. Schairer, B. Huhle, P. Vorst, A. Schilling, and W. Straßer, "Visual mapping with uncertainty for correspondence-free localization using Gaussian process regression," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[20] R. Gomez-Ojeda and J. González-Jiménez, "Robust Stereo Visual Odometry through a Probabilistic Combination of Points and Line Segments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[21] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics & Automation Magazine (RAM)*, 2011.

[22] C. E. Rasmussen, *Gaussian processes for machine learning*, 2006.

[23] S. Maneewongvatana and D. M. Mount, "It's okay to be skinny, if your friends are fat," *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*, 1999.

[24] D. Fox, "KLD-sampling: Adaptive particle filters," in *Advances in neural information processing systems (NIPS)*, 2001.

[25] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Visual Instance Retrieval with Deep Convolutional Networks," *CoRR*, 2014.