The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Learning to Localize with Gaussian Process Regression on Omnidirectional Image Data

Benjamin Huhle, Timo Schairer, Andreas Schilling and Wolfgang Straßer

Abstract— **We present a probabilistic localization and orientation estimation method for mobile agents equipped with omnidirectional vision. In our appearance-based framework, a scene is learned in an offline step by modeling the variation of the image energy in the frequency domain via Gaussian process regression. The metric localization of novel views is then solved by maximizing the joint predictive probability of the Gaussian processes using a particle filter which allows to incorporate a motion model in the prediction step. Based on the position estimate, a synthetic view is generated and used as a reference for the orientation estimation which is also performed in the Fourier space. Using real as well as virtual data, we show that this framework allows for robust localization in 2D and 3D scenes based on very low resolution images and with competitive computational load.**

I. Introduction

Omnidirectional cameras are very popular sensors for various kinds of mobile agents. Visual sensors are necessary for many tasks involving interaction with the environment as well as for tele-operation. It is therefore desirable to use visual information also for localization. In previous work, various methods have been developed that solely rely on image data in order to determine the relative position of the camera, *i.e.*, the mobile agent, and simultaneously build a model of the environment (*e.g.* [1], [2]). This is known as *structure-from-motion* in the vision community and is closely related to the problem of SLAM. The work of Nistér et al. [3] also relies on an implicit geometric modeling of the scene, yet focusing on the ego-motion estimation.

In contrast, we deal with the problem of localization in a known environment, where we estimate the position and orientation of the mobile agent. In general, this environment can be given as a metric map or as in our case, as a database of sensor measurements with associated metric pose information, in contrast to purely topological approaches, where the associated information is of semantic nature.

First, in an offline step, we record panoramic images at known positions. Using a frequency domain representation of the omnidirectional images not only leads to a compact description of the image data but also allows us to use the *energy* of the signal as a similarity measure that is invariant under rotations.

We use the statistical framework of Gaussian process regression to model the variation of the image energy in the environment, where we train the model with the prerecorded images. Since the image energy is computed for different frequency bands independently, we use multiple Gaussian processes to account for the multivariate outputs of the model. To solve the localization problem, we apply a particle filter and use the joint predictive probability of the Gaussian process model to rate the particles based on their positions and the actual view.

The same model is used to synthesize a reference view from which we estimate the orientation of the mobile agent with regard to the most likely position. This is performed in Fourier space as well, using normalized cross-correlation as similarity measure and applying a second particle filter to solve for the most likely orientation.

In the following subsection we shortly review relevant previous work and in Section II the techniques employed in our algorithm are introduced. Based upon this, Section III describes our novel methods for position and orientation estimation. Results from several experiments using virtual as well as real data sets are presented in Section IV, followed by a conclusion in Section V.

A. Related Work

In many existing approaches, the entry in the database which is most similar to the new measurement is retrieved and its associated position is taken as a hypothesis for the current location. For this task, various different image signatures and similarity measures have been proposed. For example, Menegatti and colleagues [4] use global image descriptors, whereas Andreasson et al. [5] employ sets of descriptors of local image features for topological, as well as for metric localization in conjunction with odometry readings [6]. Similarly, but purely image-based, Sim and Dudek [7] solve this problem by using generative models of the appearance of local feature neighborhoods with regard to varying viewpoints.

Omnidirectional images can efficiently be represented in their spherical harmonics basis, *i.e.* as Fourier transforms on the sphere. Friedrich et al. [8], [9] use these descriptors to localize a robot in 2D. They search for the nearest neighbor in terms of the L_2-norm of the coefficients on an interpolated grid of reference views. Contrarily, we perform regression in the frequency domain, using Gaussian processes that allow for a metric localization. On other modalities, Gaussian processes have been used for metric localization before. Schwaighofer et al. [10] apply this framework in order to localize cell phones, based on measurements of the signal-strength from multiple base stations. Similarly, Ferris and colleagues [11] used Gaussian process latent variable models for SLAM in WiFi networks.

The authors are with the Department of Graphical Interactive Systems WSI/GRIS, University of Tübingen, Germany.
{huhle,schairer}@gris.uni-tuebingen.de

II. Preliminaries

A. Harmonic Analysis on the Sphere

Omnidirectional images can be considered as a function $f(\theta, \phi) = f(\omega)$ on the 2-sphere, where $\theta \in [0, \pi]$ denotes the colatitude and $\phi \in [0, 2\pi)$ denotes the azimuth. Driscoll and Healy [12] showed that the spherical harmonic functions Y_l^m form a complete orthonormal basis over the unit sphere and that any square-integrable function $f \in L^2(S^2)$ can be expanded as a linear combination of spherical harmonic functions (Spherical Fourier Transform, SFT)

$$f(\omega) = \sum_{l \in \mathbb{N}} \sum_{m \in \mathbb{Z}, |m| \leq l} \hat{f}_l^m Y_l^m(\omega), \tag{1}$$

where $\hat{f}_l^m \in \mathbb{C}$ are the complex expansion coefficients. The spherical harmonic function Y_l^m of degree l and order m is given by

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) \exp(im\phi), \tag{2}$$

with P_l^m denoting the associated Legendre polynomials. Our input data, the spherical functions f, are defined on a uniformly sampled equiangular grid. A perfect reconstruction from a $2B \times 2B$ grid is possible when bandlimiting f to B.

Since rotating a spherical function does not mix coefficients of different degrees l, *i.e.*, of different frequency-bands, the norms of these subgroups of coefficients are invariant under arbitrary 3D rotations of the signal [13]. Therefore, the bandwise L_2-norms can be considered as a kind of energy spectrum $\mathbf{e} = (e_1, ..., e_B)^T$, where

$$e_l = \sqrt{\sum_{|m| \leq l} |\hat{f}_l^m|^2}. \tag{3}$$

This compact, rotationally invariant representation can be used when comparing pairs of spherical signals.

Kostelec and Rockmore [14] presented a method to estimate the alignment of images defined on the sphere using cross-correlation as a similarity measure. They showed that the correlation between two images g and h as a function

$$C(R) = \int_{S^2} g(\omega)\, \Lambda(R) h(\omega)\, d\omega \tag{4}$$

of rotations can efficiently be evaluated in the Fourier domain. Here, Λ denotes the rotation operator corresponding to the rotation $R = R(\alpha, \beta, \gamma)$ where α, β, γ are the Euler angles (in *ZYZ* representation) defining the rotation. Further, the spherical harmonic functions Y_l^m form an orthonormal basis for the representations of $SO(3)$ and the $SO(3)$ Fourier transform (*SOFT*) coefficients of the correlation of two spherical functions can be obtained directly by calculating the bandwise outer product (denoted by $\diamond$) of their individual SFT coefficients. Taking the inverse *SOFT*,

$$C(R) = \text{SOFT}^{-1}\left(\hat{g} \diamond (\hat{h})^\star\right), \tag{5}$$

where $(\hat{h})^\star$ denotes the complex conjugate of $\hat{h}$, yields the correlation $C(R)$ evaluated on the $2B \times 2B \times 2B$ grid of Euler angles G and its maximum value ideally indicates the rotation separating the two images. The accuracy of the rotation estimate $\tilde{R} = \arg\max_{(\alpha,\beta,\gamma) \in G} C(R(\alpha, \beta, \gamma))$ is directly related to the resolution of the likelihood grid which in turn is specified by the number of bands used in the SFT. Given images of bandwidth B, the resolution of the likelihood grid implicates an inaccuracy of up to $\pm(\frac{180}{2B})^\circ$ in α and γ, and $\pm(\frac{90}{2B})^\circ$ in β. The cubic computational cost when evaluating the grid, in practice, restricts this method to bandwidths up to $B = 256$.

When acquiring omnidirectional images, typically, the sensors do not cover the whole sphere and the images have limited support. In our previous work [15] we show that the spatially normalized cross-correlation (NCC) of two spherical images can be expanded in terms of simple correlations and therefore can be computed with multiple applications of the inverse *SOFT* transform. In the remainder of the paper we use this function as a similarity measure when estimating the orientation.

B. Gaussian Process Regression

Gaussian processes provide an elegant and powerful framework for probabilistic regression. We very briefly review the basics of Gaussian process regression. For further details, please refer to the comprehensive treatise by Rasmussen and Williams [16].

A Gaussian process (GP) can be used to model the underlying function $\tilde{f}$ of a set of observations

$$y_i = \tilde{f}(x_i) + \epsilon, \quad i = 1, \ldots, m, \quad \tilde{f} : \mathbb{R}^D \to \mathbb{R}, \tag{6}$$

that are corrupted by white Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$. In Gaussian process regression, one estimates a posterior distribution over functions f incorporating the training data $D = \{(x_i, y_i)\}_{i=1}^m$. Assuming that function values at different positions x_i are correlated, one models the covariance of the function values as a function of the inputs:

$$cov\left(f(x_i), f(x_j)\right) = k(x_i, x_j). \tag{7}$$

The posterior process $\mathcal{F}$, which models the predictive distribution $p(y|x^*, D)$, is completely specified by its mean and covariance functions. Denoting the vector $[k(x_1, x^*), \ldots, k(x_m, x^*)]^T$ by $\mathbf{k}(x^*)$ and $(y_1, \ldots, y_m)^T$ by $\mathbf{y}$, for a new position x^*, we can derive these functions as [16]

$$\mu(x^*) = \mathbf{k}(x^*)^T K^{-1} \mathbf{y}, \tag{8}$$

and

$$\sigma^2(x^*) = k(x^*, x^*) - \mathbf{k}(x^*) K^{-1} \mathbf{k}(x^*), \tag{9}$$

where K is the $m \times m$ data covariance function, such that $K_{ij} = k(x_i, x_j) + \delta_{ij}\sigma_n^2$.

The model selection and learning consist of choosing a suitable covariance function k and adjusting the so-called hyperparameters Θ. Consistently, we experienced the best

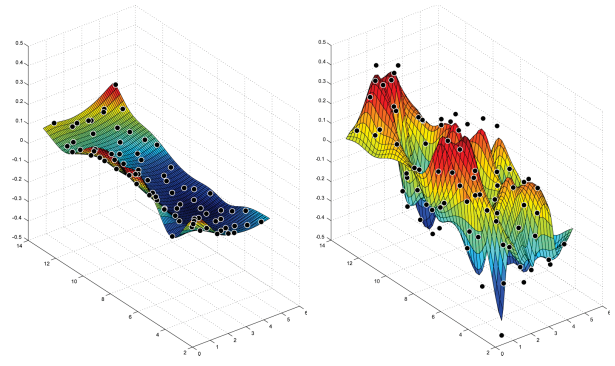


Fig. 1. Gaussian process models of the energy coefficients e_1 (left) and e_{11} (right) of the "Virtual Lab (2D)" scene with values of the reference samples depicted by black nodes. Note the more approximative nature at the higher degree.

results with a Matérn covariance function (of fixed smoothness parameter $\nu = \frac{5}{2}$, according to [16]),

$$k(x_i, x_j) = k(r) = \sigma_s \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp\left(-\sqrt{5}r\right), \tag{10}$$

where $r = \|M^{-1}(x_i - x_j)\|_2$ and the characteristic length-scale matrix is given by $M = diag(\ell)$. This covariance function implements automatic relevance determination (ARD, [17]), *i.e.*, it determines the scaling and accordingly the relevance of each dimension independently.

To adjust the hyperparameters $\Theta = (\ell, \sigma_s, \sigma_n)$, which identify the *length-scale* ℓ, the *signal variance* σ_s and the *prediction noise* σ_n, we use the marginal likelihood $\log p(\mathbf{y}|D, \Theta)$ as the optimization criterion.

III. Localization

In this section, we describe the proposed localization algorithm. Additionally, a short summary is given in Table I.

A. Offline Learning Stage

To localize a mobile agent in a given environment, we learn the appearance variation in an offline training stage. Therefore, we record a set of images $\mathcal{I}$ at known positions $\mathcal{X} = \{x_k\}_{k=1,\dots,m}$ and with orientations $\mathcal{Q}$. These images are stored as compact SFT coefficient vectors (we use bandwidths up to $B = 64$) along with their poses. Additionally, the set of rotationally invariant energy vectors $\mathcal{E} = \{\mathbf{e}^{(k)}\}_{k=1,\dots,m}$ is precomputed. For each degree l of the energy representation, we independently learn a Gaussian process $\mathcal{F}_l$ as the generative model $p(e_l|x^*, \mathcal{X}, \mathcal{E})$, *i.e.*, we learn a mapping from position x to energy vector coefficient e_l. In Figure 1, examples of learned models for different energy coefficients are shown, where for better understanding we used a 2D scene. The adaptation of the hyperparameters is done using gradient descent on the negative log marginal likelihood.

B. Position Estimation

Due to the rotationally invariant energy representation of the images, it is possible to estimate the position in a first step, independently of the orientation. The model of the appearance variation consisting of the Gaussian processes $\mathcal{F}_l, l = 1, \dots, B$ allows us to compute the probability of a measurement $\tilde{\mathbf{e}}$ conditioned on the position. Assuming statistical independence across the entries of the energy vectors, we get

$$\begin{aligned} p(\mathbf{e}|x^*, \mathcal{X}, \mathcal{E}) &= \prod_{l=1,\dots,B} p(e_l|x^*, \mathcal{X}, \mathcal{E}) \\ &= \prod_{l=1,\dots,B} \mathcal{N}\left(e_l; \mu_l(x^*), \sigma_l^2(x^*)\right). \end{aligned} \tag{11}$$

Since we assume a smooth trajectory of the mobile agent, we use a particle filter approach to estimate the most likely position $\hat{x}^{(t)}$ at each time step t conditioned on all previous measurements $\tilde{\mathbf{e}}^{(1:t)}$,

$$\hat{x}^{(t)} = \arg\max\ p\left(x^{(t)}|\tilde{\mathbf{e}}^{(1:t)}\right). \tag{12}$$

Using the *Sampling Importance Resampling* (SIR) algorithm [18], this is accomplished computing the weight update according to Equation 11 and accounting for the motion model in the prediction step by drawing new positions $x_i^{(t)} \sim p\left(x^{(t)}|x_i^{(t-1)}\right)$ for all particles $p_i,\ i = 1, \dots, n$.

C. Orientation Estimation

The orientation of the mobile agent at a given time t can be determined using the technique described in Section II-A. Namely, the normalized cross correlation between the actual view and a reference view is evaluated for a grid of possible rotations using SOFT. As the reference view, one could choose one of the prerecorded image samples of the set $\mathcal{I}$, *e.g.*, the nearest neighbor. However, in general, these samples are not dense enough, so that a robust orientation estimate is infeasible since the image content can vary significantly due to the translational offset.

Therefore, we use the generative model from Equation 11 to synthesize an SFT vector ${}^*\hat{\mathbf{f}}$ that resembles the appearance at this position. Given the most likely position $\hat{x}^{(t)}$ of the mobile agent and extending Equation 8 to the set of Gaussian processes $\{\mathcal{F}_l\}_{l=1,\dots,B}$, an energy vector can be synthesized:

$$\hat{\mathbf{e}} = \begin{bmatrix} \hat{e}_1 \\ \vdots \\ \hat{e}_B \end{bmatrix} = \begin{bmatrix} \mu_1(\hat{x}^{(t)}) \\ \vdots \\ \mu_B(\hat{x}^{(t)}) \end{bmatrix} = \begin{bmatrix} \mathbf{k}_1(x^*)^T K_1^{-1} E_1 \\ \vdots \\ \mathbf{k}_B(x^*)^T K_B^{-1} E_B \end{bmatrix}, \tag{13}$$

where E_l denotes the vector $(e_l^{(1)}, \dots, e_l^{(m)})^T$. This can also be formulated as a linear combination of the training samples,

$$\hat{e}_l = \sum_{k=1,\dots,m} a_l^{(k)} e_l^{(k)}, \tag{14}$$

with $\mathbf{a}_l = (a_l^{(1)}, \dots, a_l^{(m)})^T = \mathbf{k}_l(x^*)^T K_l^{-1}$, for every degree l. The same weights can be used to synthesize the elements of the SFT vector ${}^*\hat{\mathbf{f}}$ as linear combinations

$${}^*\hat{f}_l^m = \sum_{k=1,\dots,m} a_l^{(k)} (\hat{f}_l^m)^{(k)}, \tag{15}$$

where the $\hat{\mathbf{f}}^{(k)}, k = 1, \dots, m$ denote the spherical harmonics representation of the training samples. This can be interpreted as an approximation of the variation of the

SFT vectors, restricting the complexity to B instead of B^2 Gaussian processes. The approximation would exactly hold if, across the scene, the $\hat{f}_l^m$ for a fixed l varied proportionally to $\|(\hat{f}_l^{-l}, \ldots, \hat{f}_l^l)^T\|_2 = e_l$.

To account for the temporal coherence and to compensate the quantization effect of the SOFT-based orientation estimates, we apply a second particle filter on these estimates as described in our previous work [19].

0. Offline Stage
- record images with associated poses
- for each degree l:
  - train Gaussian process $\mathcal{F}_l$ as a mapping from x to e_l

1. Online Localization
- loop:
  - acquire new image
  - transform to Fourier space (Eq. 1)
  - compute energy vector $\mathbf{e}$ (Eq. 3)
  - position estimation
    - predict particle filter positions $x_i^{(t)}$
    - for each particle p_i
      - compute weight $w_i^{(t)} = p(\mathbf{e}|x_i^{(t)}, \mathcal{X}, \mathcal{E})$ (Eq. 11)
    - output position $\hat{x}$ of particle with maximum weight
    - resample particles
  - orientation estimation
    - synthesize ${}^*\hat{\mathbf{f}}$ for given position $\hat{x}$ (Eq. 15)
    - compute *NCC grid* of actual view and ${}^*\hat{\mathbf{f}}$ (cf. [19])
    - apply particle filter as in [19]

TABLE I
Summary of the localization algorithm.

IV. Experiments and Results

To evaluate the performance of our proposed localization technique, we conducted experiments on virtual as well as real data sets of different scales using different combinations of reference views and bandwidths and present the results in terms of root mean squared error (RMSE) in Table II. Our prototype implementation is realized in Matlab and makes use of the GPML toolbox published with [16] and of MEX-files as interfaces to C-routines of the S2Kit [20] and the SOFT library [14]. For numerical reasons we normalize the dimensions of the energy vectors such that $\mathbf{e}^{(1)} = \mathbb{1}$ and center the set of training outputs $\mathcal{E}_l$ for each Gaussian process $\mathcal{F}_l$. As reference we tried to localize by weighting the particles with the distance to the nearest neighbor reference sample in terms of energy similarity. However, this weighting scheme performed very poorly, therefore no quantitative results are included in the table. We used 400 particles for the two-dimensional and 600 particles for the three-dimensional position estimation with particles initialized at random positions to account for a "kidnapped robot" scenario. Orientation estimation was performed according to [19] using grid-based sensing with 500 particles followed by averaging in the exponential chart without further numerical optimization. Note, that using a regular grid of reference views leads to poor results when estimating the position, since the learning of the covariance functions is affected by aliasing. Therefore, we sampled the scene randomly which leads to a better distributed set of inter-sample distances r which is used to learn the covariance function.

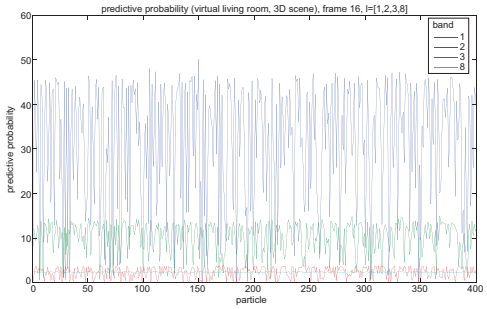


Fig. 3. Contributions of the single bandwise models $p(e_l|x^*, \mathcal{X}, \mathcal{E})$, $l = 1, 2, 3, 8$ to the joint predictive probability (Eq. 11).

A. Virtual Scenes

To acquire photorealistic image data along with ground truth position information we created omnidirectional views from CAD models using a rendering software based on global illumination. We evaluated the performance on two virtual scenes, one used for localization in 2D ("virtual lab") that is challenging due to strong symmetries in image content and a second one for 3D localization ("virtual living room"). The estimated trajectory can be seen in Figure 2a and 2b.

B. Real Scenes

We acquired omnidirectional images at approx. one frame per second using a LadyBug2 spherical camera system and recorded the reference poses as well as the trajectory of the camera with an optical motion capture system [21]. Note, that the system does not provide explicit error bounds, however, the data is evidently not noisy and typically measurements performed with such systems are precise in relation to the expectable self-localization performance. All recordings were performed under similar lighting conditions. Differing illumination could be compensated for to a certain degree by normalizing the SFT vectors with regard to their overall length. Note, that in the 2D case the camera was moved at fixed height above ground, but no additional prior information was taken into account to restrict the position estimates to a plane or to limit the rotation estimation to two dimensions, respectively. Plots depicting the estimated trajectory can be seen in Figure 2c and 2d.

C. Discussion

The results show that our localization technique leads to robust and accurate localization estimates in two and three dimensions for both virtual and real scenes even for very low resolution images (a bandwidth of, *e.g.*, 16 corresponds to panoramic images of size 32×32 pixels). It can be observed in all scenes that using less reference views, the position estimation quality slightly degrades. Due to the lower sampling density, energy coefficients for higher degrees are less distinctive. Using higher bandwidths, however,

Scene	# reference views	# frames	Bandwidth	RMSE $[m]$	RMSE $[deg]$	runtime/frame $[sec]$
Virtual Lab (2D)	36	200	8	0.58	6.33	0.32
$4.5m \times 11.0m \times 3.7m$	36	200	16	**0.56**	**3.77**	0.56
	36	200	32	0.65	4.48	2.13
	78	200	8	0.18	5.38	0.38
	78	200	16	**0.17**	**3.10**	0.68
	78	200	32	0.22	3.89	2.52
	78	200	64	0.24	4.85	14.67
Virtual Living Room (3D)	64	200	8	**0.26**	5.58	0.57
$6.0m \times 8.0m \times 3.0m$	64	200	16	0.32	**4.02**	0.73
	64	200	32	0.30	4.84	2.77
	64	200	64	0.70	7.60	15.08
	125	200	8	0.16	5.30	0.51
	125	200	16	**0.14**	**3.22**	1.07
	125	200	32	0.15	4.18	2.96
	125	200	64	0.16	5.15	17.29
Real Scene (2D)	50	100	8	0.04	5.99	0.40
$5.0m \times 5.2m \times 2.8m$	50	100	16	**0.03**	3.75	0.81
(capture volume: $1.1m \times 0.7m$)	50	100	32	0.04	**3.41**	2.41
	50	100	64	0.05	4.04	15.08
	100	100	8	**0.03**	5.46	0.50
	100	100	16	**0.03**	3.96	0.89
	100	100	32	0.04	**3.30**	2.78
	100	100	64	0.05	3.88	15.89
Real Scene (3D)	100	50	8	0.12	13.68	0.65
$5.0m \times 5.2m \times 2.8m$	100	50	16	**0.10**	10.98	1.04
(capture volume: $1.6m \times 2.0m \times 1.3m$)	100	50	32	0.14	11.01	3.07
	100	50	64	0.15	**10.54**	16.01
	200	50	8	0.12	12.63	0.89
	200	50	16	0.10	9.58	1.72
	200	50	32	**0.08**	**9.00**	4.43
	200	50	64	0.13	9.25	19.26

TABLE II

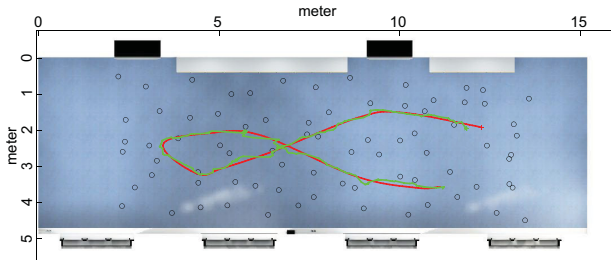
Standard Error (RMSE) w.r.t. ground-truth (virtual scenes) and motion capture data (real scenes). The runtime is measured using a non-optimized Matlab implementation on a standard quad core machine. The restricted capture volume is due to the motion capture setup.

does not necessarily improve the results, since the number of reference views and in turn the sampling density might be too low to yield distinctive energy coefficients for the upper bands and overfitting might occur. This effect can be seen, *e.g.*, when examining the results of the real 3D scene, where the best position estimation performance is achieved using a bandwidth of 16 for 100 reference views, whereas when using 200 reference views the optimum is achieved using 32 bands. Figure 3 exemplarily shows the different contributions of the single bandwise models to the joint predictive probability. This explains the fact that even for small bandwidths the model is distinctive and we obtain a good localization performance since most information is kept in the lower band models. Naturally, the accuracy of the orientation estimation improves when using higher bandwidths but is limited by the quality of the synthesized energy vector ${}^*\hat{\mathbf{f}}$ that depends on both the sampling density and the accuracy of the position estimation. A video showing the results of the different experiments is included in the proceedings and can also be found at `http://www.gris.uni-tuebingen.de/people/staff/huhle/iros2010`.

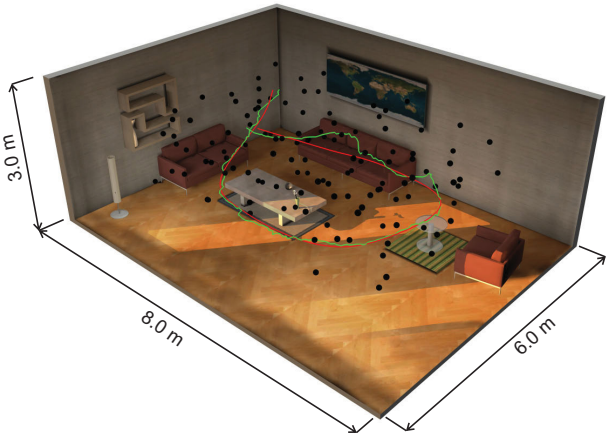
V. Conclusions

We have presented a novel method to localize a mobile agent comparing an omnidirectional image taken at its current location with a precomputed model of the variation of the image energy across the scene in different frequency bands. As image representation we use the spherical harmonics coefficients and their bandwise norms that provide a rotationally invariant description of the image. This allows us to estimate the location independently of the orientation and estimate the latter in a second stage that makes use of the location hypothesis and the same model of the image variation across the scene. This model is built using Gaussian process regression, where for each of the frequency bands a separate process is trained independently. For both the position and orientation estimation, we feed the likelihood estimates into separate particle filters. This allows us to integrate a motion model and to solve for the most likely state according to their posterior distributions.

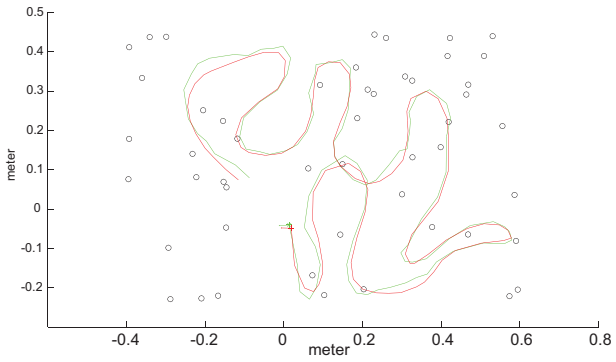
Due to the statistical learning approach, the whole system does not depend on any user-specified parameters apart from the motion model. Therefore, it can be applied to various different scenes without any modifications. We have investigated the performance of the system and experienced very robust and precise estimates in virtual as well as real scenes of two and three dimensions, with orientation varying in three degrees of freedom. The performance has proven to decay extremely slowly with decreasing resolution of the input images. This fact renders the method especially interesting for scenarios where only low resolution images are available.



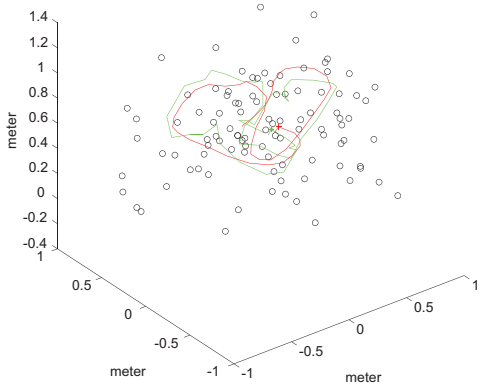
(a) **Virtual Lab (2D)**: 78 reference views and 200 test frames.



(b) **Virtual Living Room (3D)**: 125 reference views and 200 test frames.



(c) **Real Scene (2D)**: 50 reference views and 100 test frames.



(d) **Real Scene (3D)**: 100 reference views and 50 test frames.

Fig. 2. Plots of the test scenes along with ground truth position information (red), estimated trajectory (green) and positions of the reference views (black circles/dots). The bandwidth was set to 16.

ACKNOWLEDGMENTS

We thank Sebastian Herholz for setting up the motion capture system and helping us with the recordings.

REFERENCES

[1] M. Pollefeys, "Self-calibration and metric 3d reconstruction from uncalibrated image sequences," Ph.D. dissertation, ESAT-PSI, K.U.Leuven, 1999.

[2] A. Makadia and K. Daniilidis, "Correspondenceless structure from motion," *Int. Journal of Computer Vision*, vol. 75, no. 3, pp. 311–327, 2007.

[3] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[4] E. Menegatti, T. Maedab, and H. Ishiguro, "Image-based memory for robot navigation using properties of omnidirectional images," *Robotics and Autonomous Systems*, vol. 47, pp. 251–267, 2004.

[5] H. Andreasson, A. Treptow, and T. Duckett, "Localization for mobile robots using panoramic vision, local features and particle filter," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005.

[6] H. Andreasson, T. Duckett, and A. J. Lilienthal, "A minimalistic approach to appearance based visual slam," *IEEE Trans. on Robotics (Special issue on Visual SLAM)*, vol. 24, no. 5, 2008.

[7] R. Sim and G. Dudek, "Learning generative models of scene features," *Int. Journal of Computer Vision*, vol. 60, pp. 45–61, 2004.

[8] H. Friedrich, D. Dederscheck, M. Mutz, and R. Mester, "View-based robot localization using illumination-invariant spherical harmonics descriptors," in *Proc. VISAPP (2)*, 2008, pp. 543–550.

[9] H. Friedrich, D. Dederscheck, E. Rosert, and R. Mester, "Optical rails: View-based point-to-point navigation using spherical harmonics," in *Proc. DAGM*, 2008.

[10] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "Gpps: A gaussian process positioning system for cellular networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[11] B. Ferris, D. Fox, and N. Lawrence, "Wifi-slam using gaussian process latent variable models," in *Proc. Int. Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[12] J. Driscoll and D. M. Healy, Jr., "Computing fourier transforms and convolutions on the 2-sphere," *Adv. Appl. Math.*, vol. 15, no. 2, pp. 202–250, 1994.

[13] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Symposium on Geometry Processing (SGP)*, 2003.

[14] P. Kostelec and D. Rockmore, "Ffts on the rotation group," Fe Institutes Working Paper Series, Tech. Rep., 2003.

[15] B. Huhle, T. Schairer, and W. Straßer, "Normalized cross-correlation using SOFT," in *Proc. Int. Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*, 2009.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[17] R. M. Neal, *Bayesian Learning for Neural Networks*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.

[18] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, February 2002.

[19] T. Schairer, B. Huhle, and W. Straßer, "Application of particle filters to vision-based orientation estimation using harmonic analysis," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[20] P. Kostelec and D. Rockmore, "S2kit: A lite version of spharmonickit," Dartmouth College, Tech. Rep., 2004.

[21] Optitrack FLEX:V100 and Rigid Body Tracking Software Library, NaturalPoint, Inc.