

3D-Based Monocular SLAM for Mobile Agents Navigating in Indoor Environments

Dejan Pangercic
Technische Universität München
dejan.pangercic@in.tum.de

Radu Bogdan Rusu
Technische Universität München
rusu@cs.tum.edu

Michael Beetz
Technische Universität München
beetz@in.tum.de

Abstract

This paper presents a novel algorithm for 3D depth estimation using a particle filter (PFDE - Particle Filter Depth Estimation) in a monocular vSLAM (Visual Simultaneous Localization and Mapping) framework. We present our implementation on an omnidirectional mobile robot equipped with a single monochrome camera and discuss: experimental results obtained in our Assistive Kitchen project and its potential in the Cognitive Factory project. A 3D spatial feature map is built using an Extended Kalman Filter state-estimator for navigation use. A new measurement model consisting of a unique combination between a ROI (Region Of Interest) feature detector and a ZNSSD (Zero-mean Normalized Sum-of-Squared Differences) descriptor is presented. The algorithm runs in realtime and can build maps for table-size volumes.

1 Introduction

1.1 Problem Statement

The utilization of robots in everyday human-activities has recently become a significant trend of research in robotics. There are several, commercially available household robots (iRobot, Anybots) that can perform basic household chores: from cleaning the room to assistance in serving food. However, all these complex tasks, are usually pre-programmed and can not deal with the high degree of uncertainties usually associated with a human-populated environment.

In our research work, we aim to develop fully cognitive environments, where sensors and actuators are embodied in the world, and mobile robots act by making use of them. The Assistive Kitchen project [2] is a distributed sensor-equipped environment, in which a B21-like robot acts as a cognitive household assistant. The robot can, due to its flexible and adaptive characteristics, pick tasks on the run and therefore serve, facilitate at

household chores and, when needed, interact with the human fully autonomously. While a complete set of the ongoing research in the Assistive Kitchen involves geometrical representation and construction of semantic maps, high and low-level robot actions planning, 3D human motion tracking, path planning, human mimics and gestures and voice recognition; we will in this paper only focus on the robot localization and mapping part using visual perception through the monocular camera. The Cognitive Factory(CF) [19] on the other side is an evolving instance of the Flexible Manufacturing System where the B21-like robot intervenes as a mobile agent. The robot back-ups installed industrial robots in order to extend their spatial reach, performs corrections in the case of the erroneous production plans and interacts with the human fully autonomously as well. Since we have a hardware and a simulation version of the CF, we would like to stress that for the time being above mentioned robot's skills only apply to the simulated version (Figure 1). Full integration of the robot in the real CF entails to our short-term future research.

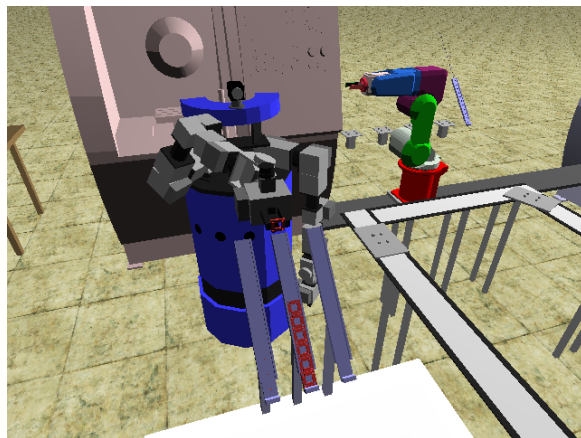


Figure 1. B21 robotic assistant filling a part feeder

The term SLAM [6, 18], refers to the problem of a

robot concurrently building a representation of the environment (i.e. map) while navigating around it and keeping itself localized. This approach accounts for any arbitrary moderate-sized indoor environment thus enabling us to turn virtually any casual indoor environment into the cognitive one.

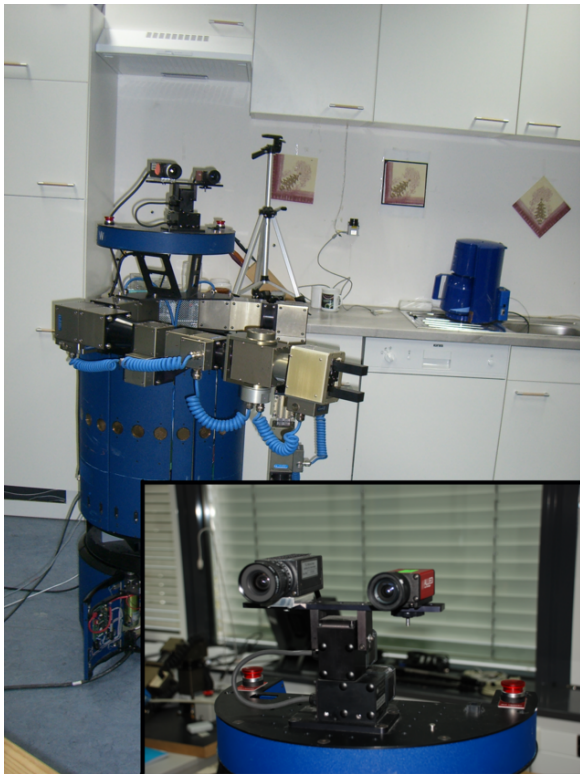


Figure 2. B21 robot with mounted Marlin camera (red) in Assistive Kitchen environment

1.2 Key Ideas

Performing SLAM based on visual perception has a number of advantages over traditional methods which deploy laser, sonar and other sensors. High-performance cameras are rather affordable nowadays due to their low-prices. They deliver high-definition-high-rate inherently intuitive images. Thanks to the Computer Vision society which has developed a broad-range of powerful and efficient image processing algorithms, one can already by default overcome certain ambiguities in feature matching. Furthermore, on account on their plurality the same camera can be used for several other applications in the context of the Assistive Kitchen, i.e. motion tracking, collision avoidance.

The elementary schema of our vSLAM algorithm was adopted from the work of Davision et al. [5] and is depicted in Figure 3. The extra-framed parts constitute our contributions and will be presented later in this paper in greater detail. The underlying idea of our system is to have an initial map knowledge (step 1), a well-fitted robot

motion model (step 2), a measurement model involving a camera (step 3 and 5) and an appropriate state-estimation tool (step 4). For the latter we selected an Extended Kalman Filter (EKF) over the runner-up candidate used in a Fast-SLAM approach [10] because we believe that localization and mapping can not be decoupled and separately estimated. The main drawback of EKF, namely its required high computational demands ($O(N^2)$), has been resolved by the careful selection and insertion of features in the SLAM map.

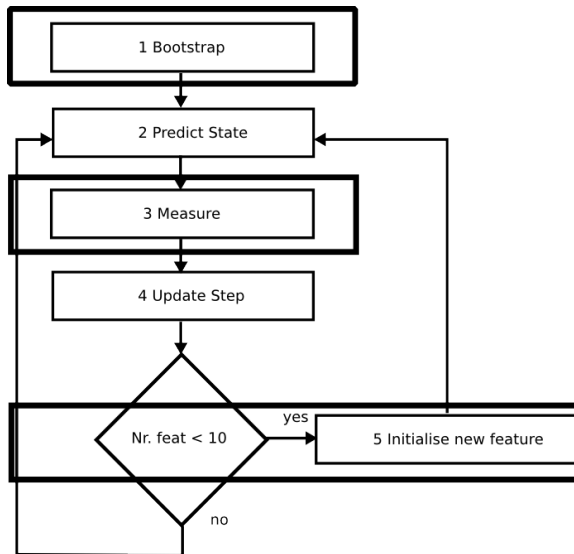


Figure 3. vSLAM algorithm flowchart

1.3 Our Contributions

In this paper, we report on our experiences while working on the vSLAM problem, focusing on the following contributions:

- a *unique measurement* model that consists of the combination between the ROI feature detector [15] and the ZNSSD [3] feature descriptor. They both demand very little computational cost while still remaining invariant to translations, rotations and scale.
- a novel technique to determine the depth of features measured from the 2D image - PFDE. The technique enables us to reliably determine the *depth* of the feature within maximum of 3 frames from when the feature was first observed. It is different from a full particle filter algorithm approach applied in [5] in that it does not perform a re-sampling step but evaluates likelihood function of all living particles, i.e. all depth hypothesis λ_n . The depth is accepted as the feature's third coordinate when a Coefficient of Variation (CV) drops below a certain threshold. CV is a measure of dispersion of a Gaussian probability distribution and is defined as the ratio of the standard deviation σ to the mean μ .

The remainder of the paper is structured as follows: Section 2 revises related work and justifies our ideas for the algorithm implementation while Section 3 presents our core work: novel techniques in the measurement model part of the vSLAM algorithm. Section 4 shows the final experimental results and Section 5 gives conclusions and future look.

2 Related Work

2.1 SLAM Overall

The most common state estimation techniques for solving a SLAM problem are [13]: Visual Odometry, FastSLAM, DP-SLAM, Graph SLAM and Extended Kalman Filter-based SLAM (EKF). We narrowed the selection for our estimator to the FastSLAM and the EKF, as the rest of the techniques either only work in a batch mode or are unable to construct long-term consistent maps.

While the FastSLAM is a novel approach, computationally more feasible and solves the full SLAM problem, the EKF is on the other side most frequently applied and a very well tested approach. The ultimate decisive point favoring the EKF was also our constraint to bootstrap the map with a certain number of *known features* which enables us to immediately place the EKF in our vSLAM algorithm. The algorithm borrows the motion model from, and is similar in the concept to the work of Davison [5], albeit it proposes a different measurement model.

2.2 Vision Perception in SLAM

In one of the first vSLAM works [12], Neira et al. presented a simple monocular system mapping vertical line segments in 2D in a constraint indoor environment. The rest of the approaches are using stereo cameras. Jung and Lacroix [9] presented a downward-looking stereo system to localize an airship. The system performs a terrain mapping and does not run in realtime mode. A work by Sim et al. [16] combines SIFTs with FastSLAM filtering and achieves fairly large-scale mapping, however it throttles processor resources by spending on average of 10 seconds per frame. Several authors, like Nister [8], presented systems based on the standard *structure from motion* methodology of frame-to-frame correspondences matches. However, these systems are unable to redetect features after the periods of neglect.

An interest aspect was explored by Sola et al. [17] where they propose to use a Bi-Camera rather than only one monocular camera or a stereo set. By this they are able to *a)* use a very well understood bearings-only measurements from the mono SLAM approach, *b)* compensate for downsides of the stereo SLAM, i.e. widen the range of landmarks' observability and *c)* refrain from relying on the stereo rig calibration. Encouraged by the possibility to upgrade to the Sola's work if needed, and the fact that stereo-based vSLAM is computationally more expensive, we decided to build our algorithm based on the *monocular* camera measurements.

3 Measurement Model

Our motion model accounts for 6 Degrees Of Freedom (DOFs) and is in fact redundant for the planar-like setups that we have (translation in $x - z$ plane, rotation around y axis in cartesian world frame). However, we have noticed that translation along, and rotations around unused axes, on average always converge to *zero* and we therefore considered them as non-perturbating factors.

The measurement part of our vSLAM algorithm asserts an initialization and matching of the features. The SLAM map is bootstrapped with 4 manually inserted, consistent features with known positions and covariances in the world coordinate frame while the rest of them are brought in the map using PFDE. Estimated 3D feature positions are projected into the 2D image over the camera perspective projection model and measured in order to yield a discrepancy (innovation) between the projected and the actual patch position [13]. In the following sections we discuss the operations performed on consistent features in the image plane, as well as the process of initialization and matching of the Newly Initialized Feature (NIF). We also present its transformation to the 3-element consistent representation using PFDE.

3.1 Feature Detector

We utilized an industrial camera type Marlin F-046C from Allied Vision Technologies, which delivers monochrome, 8-bit, 320×240 pixels images.

Distinctive features in the image were detected by the ROI operator which is elementary similar to a Harris Corner Detector (HCD) [7]. Second-order derivatives g_{xx} , g_{yy} , g_{xy} in x , y , and xy directions over a patch of the image (we use a 11×11 pixels patch) are calculated, and a 2×2 symmetric covariance matrix D is built.

By performing eigenanalysis, the *least* of the two eigenvalues e of the given covariance matrix D are checked against an eigenvalue magnitude threshold ($threshold < max(e)$). If this threshold condition is fulfilled, the feature with the *largest* eigenvalue magnitude is deemed salient and gets extracted.

$$D = \begin{bmatrix} g_{xx} & g_{xy} \\ g_{yx} & g_{yy} \end{bmatrix} \quad (1)$$

The search for new features is always performed in regions of the image that do not contain any features yet. A search box of approximately 80×60 pixels is randomly placed and convoluted for the patch with the highest eigenvalue magnitude. Figure 4 shows the magnitudes of the least of two eigenvalues per feature of all the features found in the corresponding cyan box.

3.2 Feature Descriptor

The second component in having a successful feature operator for tracking is its *re-detection* part, and is by far the most complex task to realize in such a framework. In

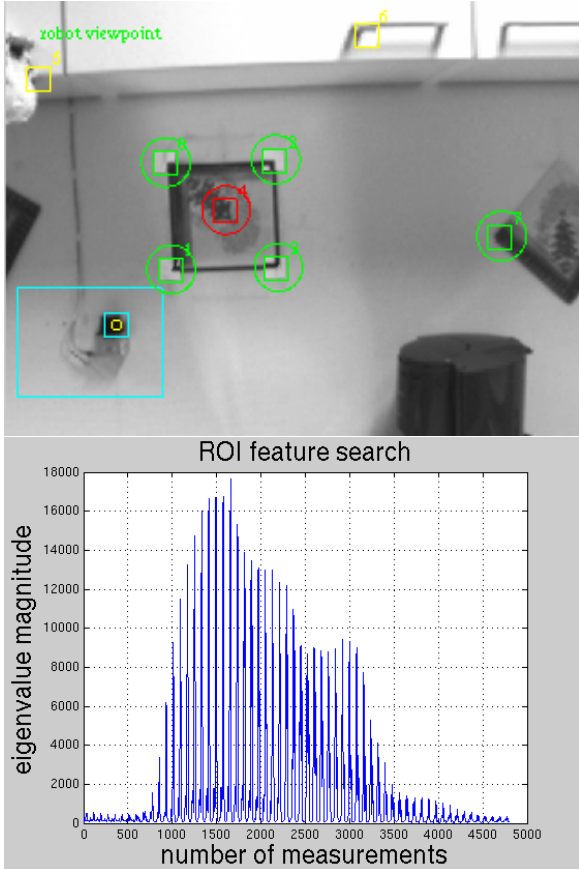


Figure 4. Top: Feature detected, measurement started (see cyan box), Bottom: Magnitudes of the least of 2 eigenvalues per feature for 4800 features found in the related 80×60 sized cyan box above

general in SLAM, this problem is termed *data association*, i.e. a problem of matching the corresponding features in successive images.

In order to resolve these ambiguities, the feature descriptor has to be translational, rotational and scale invariant. We will show that the ZNSSD criteria [3] that we used, performs in an invariant manner under moderately small robot motion.

$$ZNSSD = \sum_{i,j}^{n^2} \left[\frac{I_{ij} - \bar{I}}{\sigma_I} - \frac{J_{ij} - \bar{J}}{\sigma_J} \right]^2 \quad (2)$$

$$C = \frac{ZNSSD}{n^2} \quad (3)$$

The symbol I in the equation 2 above denotes an illuminance value of the patch in the image that was initially observed, and J symbolizes an illuminance value of the patch that is currently being measured. \bar{I}, \bar{J} are patches' mean illuminances whereas σ s represent standard deviations. A patch, in our case, has a size of 11×11 pixels which represents a balance between accuracy and computational time. Subscripts i,j refer to the pixel location

inside the patch. Equation 2 computes the sum-of-squared differences between those two patches and makes a *zero-mean*, while the final normalized correlation C is obtained by dividing this equation by the number of pixels (n^2) in one patch (Equation 3).

After performing several tests with various *correlation thresholds* C , we have selected $C = 0.4$, i.e. in case that C between the initial and the current patch overcomes this value, the measurement of the current feature is marked as failed. Figure 5 top presents the details for frame number 525 from one frame sequence, where all measurements for the selected features (except number 20 – red) returned at least one value under the C threshold. The least 10 correlation values per selected feature measurement are shown in the bottom part of Figure 5. Figure 6 shows

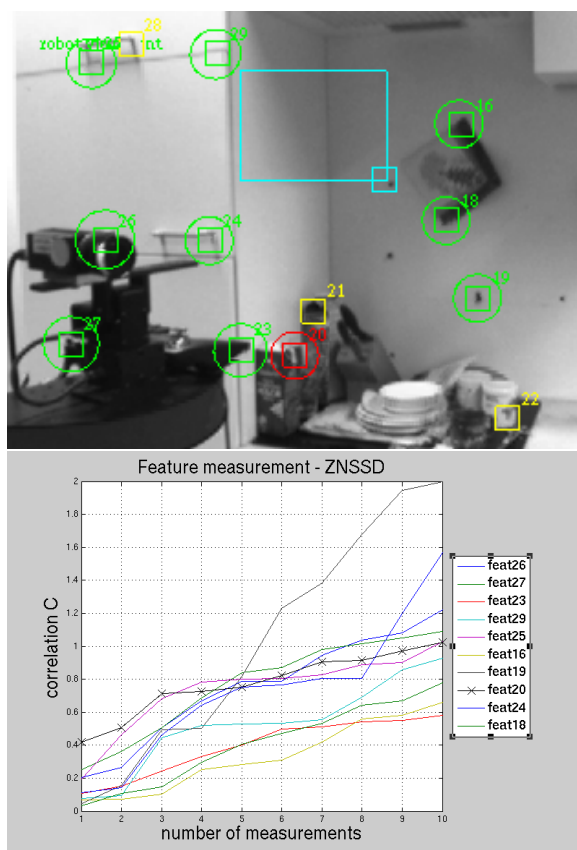


Figure 5. Top: Detail in frame 525, feature 20 measurement failed, Bottom: Least 10 correlation values for feature measurements in the upper image. Note: Feature Nr. 20 never undershoots $C = 0.4$ and is deemed as false re-detected

that ZNSSD is in fact translational and rotational invariant. The topmost image shows the situation prior to the SLAM building. In the middle, the situation when the robot was translated $1m$ backward on world frame z -axis is shown. In the very bottom the robot(camera) was rotated -30° around the world frame z -axis and the image certifies that ZNSSD is to the extent, also rotational invariant. We did not perform any particular tests to assess

the scale invariance, since the top and the middle image in Figure 6 clearly result in scale change, however the features are still being reliably re-detected.



Figure 6. Top: Initial setup, robot 0.7m away from origin, **Middle:** Robot was translated 1m backward on z-axis, **Bottom:** Robot(camera) was rotated -30° around z-axis. The world coordinate frame is in the center of the rectangle with x-left, y-up, z-forward

3.3 Particle Filter for Depth Estimation - PFDE

When using a monocular camera in vision applications, we commonly have one serious drawback: a lack of disparity and thus a lack of the depth coordinate. However, the problem is solvable if we can observe the same landmark from many different poses. Currently, the two main approaches to solve this problem are a Unified Inverse Depth Parameterization [11] and a 1D Particle Filter Depth Estimation (PFDE) [4]. Both approaches propose a 6-element vector for the NIF, where the one that we adopted looks like (see Eq. 5 and 6 for the explanation of subscripts):

$$y = (y_{Xl}^W \quad y_{Yl}^W \quad y_{Zl}^W \quad y_{Xd}^W \quad y_{Yd}^W \quad y_{Zd}^W)^T. \quad (4)$$

The key idea of the *modified* (our contribution) PFDE is that the full 3D coordinates can not be directly retrieved from the 2D ones, but one can always unproject a directional normalization vector pointing towards the 3D feature location along the camera projection ray. The projection ray can be imagined as a line in the 3D world space with one end at the camera projection center and the other end somewhere in the “*semi-infinity*” (see Figure 7 - top). The feature being sought must lie somewhere in the 3D space along that line. By “*semi-infinity*” we actually refer to long distances that are appropriate for indoor environments, i.e. in our case the maximum line length is set at 5m.

NIF’s 6-element vector is constructed by extracting the robot’s 3D position coordinates (x_p) in world frame at the time the feature was first observed and thus obtaining the first 3 elements (l subscript denotes line’s one end coordinates):

$$\begin{pmatrix} y_{Xl}^W \\ y_{Yl}^W \\ y_{Zl}^W \end{pmatrix} = \begin{pmatrix} x_p(0) \\ x_p(1) \\ x_p(2) \end{pmatrix}. \quad (5)$$

The second 3-element partition is less trivial, it requires a *camera perspective unprojection model* [13] which projects a 2D image point into homogeneous coordinates. Furthermore, such obtained coordinates have to be normalized and pre-multiplied with the robot rotation in the world coordinate frame ($Rot(q^{W-R})$) in order to yield a direction in the world frame:

$$\begin{pmatrix} y_{Xd}^W \\ y_{Yd}^W \\ y_{Zd}^W \end{pmatrix} = Rot(q^{W-R}) * \sim \quad (6)$$

$$\sim norm \left[\begin{pmatrix} \frac{(Iud-u_0) \cdot s_x}{kd((Iud-u_0)^2 s_x^2 + (Ivd-v_0)^2 s_y^2) + 1} \frac{-1}{f} \\ \frac{(Ivd-v_0) \cdot s_y}{kd((Iud-u_0)^2 s_x^2 + (Ivd-v_0)^2 s_y^2) + 1} \frac{-1}{f} \\ 1.0 \end{pmatrix} \right].$$

The unknown variables in the above equation are: f - focal length, Iud, Ivd - distorted image coordinates, u_0, v_0 - optical center in image coordinates, s_x, s_y - horizontal and vertical pixel sizes on the CCD sensor, and kd - the 1st radial distortion coefficient.

Once the NIF vector is obtained, we insert it in the SLAM map as well, let it run through the EKF and apply PFDE on it. Since the camera projection ray is populated with N initially uniformly distributed, 1D particles (Figure 7 - bottom), where each represents a *hypothesis* of the currently processed feature’s depth λ_n , the measurement of NIF is far more complex than that of the consistent 3-element feature. First, we have to transform NIF to the robot coordinate frame (zeroing) and coincide it with the world frame. However, since we cope with a 6-element vector, we have to proceed in two steps, first we zero the end of the projection ray (Equation 7), and then the direction coordinates (Equation 8). Please note that PFDE does not draw particles randomly, but draws them all in a *sequential manner*. Each particle’s weight π_0^n initially

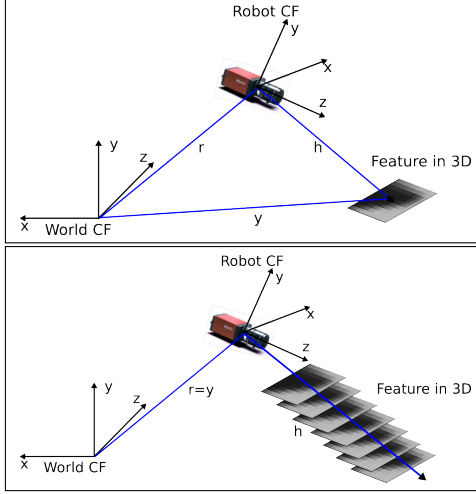


Figure 7. Top: System geometry, r - robot (camera) world frame position, y - feature world frame position, h - feature position in robot frame zeroed to world frame, **Bottom:** populated camera projection ray

equals $\frac{1}{N} = p_0(\lambda_n)$.

$$h_i^R = Rot((q^{W-R})^{-1}) \cdot (y_i^W - r^W) \quad (7)$$

$$h_d^R = Rot((q^{W-R})^{-1}) \cdot y_d^W \quad (8)$$

In the next step we have to calculate N possible 3D features for the N number of λ s using:

$$\overline{h}_n^R = h_i^R + \lambda_n \cdot h_d^R. \quad (9)$$

We opt to put “hat” on the variable h_n^R denoting a possible 3D feature in the robot frame as it is important later on when calculating the Jacobians to appropriately propagate the covariances.

Once we have the set of the 3D features along the projection ray in the robot frame, we are able to predict measurements by normally applying the camera perspective projection model.

This set of features results in a set of N overlapping ellipses in the 2D image after the projection is done. Since this set of ellipses has to be convoluted to find the matching patch, we had to find a way to computationally ease the problem. The solution to solve it lies in setting up an auxiliary image with the same size as the one we are processing (320×240 pixels) and setting all pixel illuminances to some impossible value, -1 for instance. In the course of the search, these values are then replaced by the correlation value that the same pixel location in the truly convoluted image returns. Every image pixel has therefore only been convoluted once per measurement.

The set of features is evaluated using the Gaussian measurement probability (Equation 10) measure as described in [18]. The outcome of the evaluation method is assigned to the new weight for the measured particle. We

select the gaussian filter for evaluation because we later use the CV criterion in order to decide whether the particle set has yielded a sufficiently small *dispersion* and can be converted to the 3-element representation or not. Finally, the particle’s probability is calculated over the Bayes’ rule by multiplying the current weight with the prior particle probability (Equation 11).

$$\pi_{k+1}^n = \frac{1}{\sqrt{2\pi s}} \cdot e^{-\frac{1}{2} \cdot \nu^T s^{-1} \nu} \quad (10)$$

$$p_{k+1}(\lambda_n) = \pi_{k+1}^n \cdot p_k(\lambda_n) \quad (11)$$

ν and s respectively denote an innovation and an innovation covariance of each feature’s depth hypothesis in the set of particles λ_n .

After every iteration of the particle algorithm, the mean and the variance of the particle set are calculated in the following way:

$$\bar{\lambda} = \sum_{n=1}^N p(\lambda_n) \cdot \lambda_n; \quad \sigma_{\lambda}^2 = \sum_{n=1}^N p(\lambda_n) \cdot \lambda_n^2 - \bar{\lambda}^2 \quad (12)$$

Figure 8 top depicts how the particle set representation through the mean and the standard deviation becomes narrower. Notice that in frame 309 particles are distributed uniformly, while in the subsequent frames (310 – 312) they have a Gaussian form. The CV in the frame 312 falls below 0.3 and the NIF can be converted to the 3-element representation.

4 Experiments

This paper presents the final localization results of the vSLAM algorithm running in the Assistive Kitchen under the variation of the motion model and the measurement model noise parameters [5][13]. We drove the robot along the V-shaped trajectory by first moving it backwards along the world coordinate frame z-axis until the verge point, and then moving it side-forwards towards the positive x-axis. The sequence of images was later on processed with different *noise parameters* which yielded different path trajectories. The graph in Figure 8 bottom shows all trajectories plotted together with the trajectory returned by the wheel odometry. Dash separated numbers from left-right in the legend denote: image noise, linear velocity and angular velocity standard deviations. We can see that larger measurement noise actually smoothes out the trajectory while small motion model noise improves the accuracy given that the wheel odometry yielded a trajectory close to the ground truth. However, the ground truth still remains to be proven as values are returned by the Player middleware which converts the number of wheel revolutions over parameterized linear equation which could be erroneous. The noise-parameters-combination $sd = 1, \sigma_a = \frac{1m}{s^2}, \sigma_{\alpha} = \frac{2rad}{s^2}$ eventually outperformed the others and is therefore deemed to converge to the optimum of the current application setup. The video of the final test is available on the following website: <http://www9.cs.tum.edu/people/pangercic/research/etfa.avi>

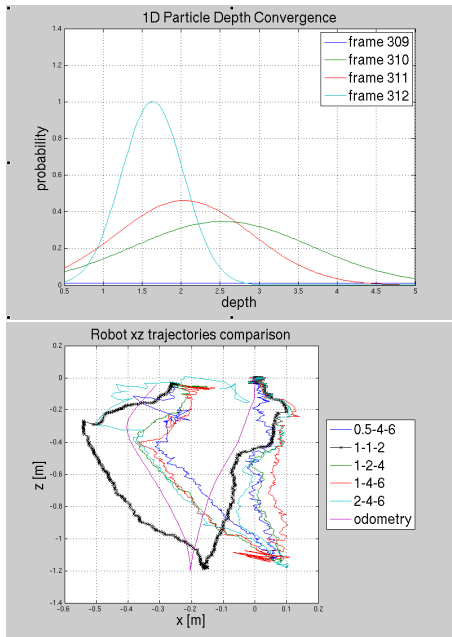


Figure 8. Top: Estimation of the feature depth using PFDE, Bottom: Variance of the noise parameters for the same sequence of images.

5 Conclusion and Future Work

The work presented in this paper addresses a *top-down* probabilistic approach for 3D SLAM using a single handheld camera attached to the omnidirectional robot operating in the Assistive Kitchen environment. The sparse, joint-covariance based map of the 11×11 pixels rich-texture features is being built during the SLAM algorithm. The map is bootstrapped by inserting four features with known correspondences whereas every next feature is extracted automatically and brought in the map over an efficient PFDE algorithm. The algorithm currently runs accurately for the table-sized volumes.

The ultimate objective of our approach is to be able to build the map of the complete indoor environment of any kind. However, to reach that we have to *a)* improve the measurement model, *b)* incorporate semantics in the map and *c)* account for the dynamic environment. The first point is amendable by implementing a better feature operator, like SURF [1] or tuples of straight 3D lines. Knowledge about objects and their positions provided by the 3D semantic maps [14] can help to find a restore pose point in case of the substantial ground-truth offset in robot trajectory. The contribution of our colleagues who work on human mimic and voice recognition and also 3D human-motion tracking will serve as a base to establish a needed level of flexibility for the dynamic environments. In particular we aim at the full integration, including the 3D-based SLAM capability, of the B21-like robot into the real and simulated instance of the Cognitive Factory. Robot's current capabilities in the latter instance are currently limited to the ground-truth odometry-based navigation, 3D laser point clouds construction, filling-in the feeders, and picking-up and placing parts on the pallets at a conveyor belt stopper.

References

- [1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV06*, pages I: 404–417, 2006.
- [2] M. Beetz, J. Bandouch, A. Kirsch, A. Maldonado, A. Müller, and R. B. Rusu. The assistive kitchen — a demonstration scenario for cognitive technical systems. In *Proceedings of the 4th COE on HAM*, 2007.
- [3] M. Cagnazzo. *Wavelet Transform and Three-Dimensional Data Compression*. PhD thesis, Università degli studi di Napoli Federico II, 2003-2004.
- [4] A. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *IEEE International Conference on Computer Vision*, pages 1403–1410, October 2003.
- [5] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007.
- [6] T. Durrant-Whyte, H.; Bailey. Simultaneous localization and mapping: part 1. *Robotics and Automation Magazine, IEEE*, 13(2):99–110, June 2006.
- [7] C. Harris and M. Stephens. A combined corner and edge detector. In *In Alvey Vision Conference*, pages 147–152, 1988.
- [8] N. D. N. O. B. J. Visual odometry. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 1:I–652–I–659 Vol.1, 27 June-2 July 2004.
- [9] I. Jung and S. Lacroix. High Resolution Terrain Mapping Using Low Altitude Aerial Stereo Imagery. In *Proc. Ninth Int'l Conf. Computer Vision*, 2003.
- [10] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using fastslam, 2003. *Proc. ICRA*, 2003.
- [11] J. Montiel, J. Civera, and A. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Robotics: Science and Systems*, August 2006.
- [12] J. Neira, M. Ribeiro, and J. Tardos. Mobile Robot Localisation and Map Building Using Monocular Vision. In *Proc. Int'l Symp. Intelligent Robotics Systems*, 1997.
- [13] D. Pangercic. Monocular 3d slam for indoor environments. Master's thesis, Technische Universität München, 2007.
- [14] R. B. Rusu, N. Blodow, Z.-C. Marton, A. Soos, and M. Beetz. Towards 3d object maps for autonomous household robots. In *Proceedings of the 20th IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2007. Accepted for publication.
- [15] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.
- [16] R. Sim, P. Elinas, M. Griffin, and J. Little. Vision-Based SLAM Using the Rao-Blackwellised Particle Filter. In *Proc. IJCAI Workshop Reasoning with Uncertainty in Robotics*, 2005.
- [17] J. Sola, A. Monin, and M. Devy. Bicamslam: Two times mono is more than stereo. In *ICRA*, pages 4795–4800. IEEE, 2007.
- [18] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, 2005.
- [19] M. F. Zäh, C. Lau, M. Wiesbeck, M. Ostgathe, and W. Vogl. Towards the cognitive factory. *Proceedings of the 2nd CARV*, 2007.