A method for discrete self-localization using image analysis

Włodzimierz Kasprzak
Warsaw University of Technology
Inst. of Control and Computation Eng.
ul. Nowowiejska 15/19, 00-665 Warsaw
W.Kasprzak@ia.pw.edu.pl

Wojciech Szynkiewicz
Warsaw University of Technology
Inst. of Control and Computation Eng.
ul. Nowowiejska 15/19, 00-665 Warsaw
W.Szynkiewicz@ia.pw.edu.pl

Abstract

A method for discrete self-localization of an autonomous mobile system was proposed. One of its many possible implementations was designed, that uses a camera subsystem, which delivers sensor information about the environment reduced to an n-elementary measurement vector. Three different algorithms of image analysis were proposed and implemented. The self-localization approach with three different image sub-systems was tested by computer simulations on different natural and synthetic scenes. A robust behavior of the approach in all cases was verified experimentally.

1 Introduction

The mobile service and diagnostic robots [2, 3] have to work in a specific "human-like" environment, that firstly prohibits the use of such active sensors, like laser scanner devices, and secondly, it requires to interact with humans, that can cross the trajectory of a moving vehicle. Both features of the environment support the use of digital cameras for the acquisition of sensory information. By using image analysis methods different tasks required for autonomous navigation can be be solved, like the detection and recognition of obstacles [6, 9], the tracking of road borders [5, 10] and the self-localization in a (partly) known environment [1, 4]. In this paper we shall propose an approach to self-localization in an indoor environment that explores the moving capability of an autonomous vehicle, i.e. which recursively adjusts its state estimation performing a repeated analysis of a sequence of images (different views) of the scene.

2 The self-localization method

2.1 The method of state condensation

As an appropriate method of discrete state estimation we choose the method of state condensation [3, 4]. It assumes, that the number of states can be limited to a finite number, i.e. specific combinations of state parameter values are 'frozen' in order to represent a particular state of the system. For a finite set of states it is computationally feasible to estimate the probability distribution of states.

Belief state – the pdf of states upon the condition of a sequence of observation:

$$\forall s^k : Bel_t(s^k) = p(s_t^k | o_t, o_{t-1}, \ldots, o_{t-n}) \qquad (1)$$

In practice a finite set of states $\boldsymbol{S}$ is defined, that covers the studied environment and during initialization of the condensation process the a priori pdf od states is specified (by default or due to a learning process):

$$\forall s^k : Bel_0(s^k) = p(s_0^k). \qquad (2)$$

The iteration cycle of the condensation algorithm consists of three main steps (1,2,4 below) and it requires an application-dependent measurement (step 3 below):

1. the prediction of belief state;

   The a priori pdf of state transition is available

   $$p(s_{t+1}^k | s_t^l, \ldots, s_0^i) = p(s^k | s^l) \qquad (3)$$

   where $s_t^l, \ldots, s_0^i$ is the history of past best belief states - the Markov criterion is assumed to be satisfied.

   On base of current belief state distribution and the above state transition one gets the prediction of next belief state, $\forall s^k$:

$$p(s_{t+1}^k | o_t, \ldots, o_0) = \sum_{s_t^i \in S} p(s_{t+1}^k | s_t^i) p(s_t^i | o_t, \ldots, o_0) \qquad (4)$$

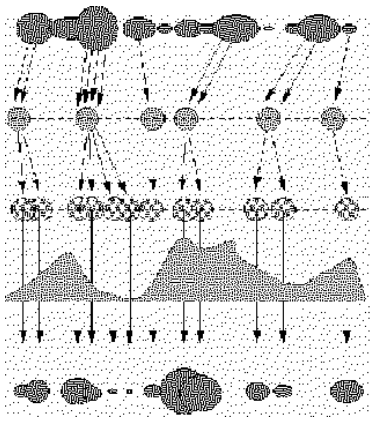


Figure 1: The state condensation scheme.

2. stochastic diffusion;

   In order to model the possible disturbances (system noise) the values of predicted belief state are spread over their neighbor states, for example:

$$p'(s^k_{t+1}|o_t,\ldots,o_0) = \sum_{s\in S}\left[p(s_{t+1}|o_t,\ldots,o_0)\frac{1}{2\pi\sigma}e^{[-\frac{1}{2}||S^k-S||^2]}\right] \quad (5)$$

3. measurement;

   The second a priori pdf should be known:
   $\forall s^k \in S : p(o|s^k)$.

4. modification of the belief state (the reaction onto the measurement).

   Let o_{t+1} is the measurement at discrete time $t+1$. The modification of belief state is finally given as:

$$\forall s^k: \quad Bel_{t+1}(s^k) = p(s^k_{t+1}|o_{t+1},o_t,\ldots,o_0) = \\ = c_{t+1}p(o_{t+1}|s^k)p'(s^k_{t+1}|o_t,\ldots,o_0), \quad (6)$$

   where c_{t+1} is a necessary coefficient for normalization of the belief state sum to 1.

2.2 The algorithm of self-localization

In autonomous navigation the action performed by the vehicle or camera are usually known, due to the odometry. Hence, this knowledge can be incorporated into the state condensation scheme. Now, the role of the learning phase is:

- for each discrete state $s \in S$ and possible measurement vector m to determine the pdf: $p(m|s)$;
- for each pair of states s^k, s^j and each possible action a to determine the pdf of state transition with respect to action: $p(s^k|s^j,a)$.

The working phase of self-localization is an extension of the state condensation scheme:

1. Get the goal state.
2. Initialization of a default belief state at $t = 0$ (for example by a uniformly distributed pdf) $Bel_0(s^k) = p(s^k_0|H_0)$.
3. REPEAT until the goal state is not reached:
   (a) $t = t + 1$;
   (b) find the current best state: s^*_{t-1} = argmax $p(s_{t-1}|H_{t-1})$, where $H_{t-1} = (s_{t-1}, m_{t-1}, s_{t-2}, m_{t-2}, \ldots, s_0, m_0)$ is the history of past belief states and measurements;
   (c) determine and perform the next action resulting from minimization of the distance between current best state and the goal state;
   (d) as the current action a_t and the a priori pdf $p(s_t|s_{t-1},a_t)$ are known the predicted belief state at time t can be computed $\widehat{Bel}_t(s^k) = \sum_s[p(s^k_t|s_{t-1},a_t)p(s_{t-1}|H_{t-1})]$;
   (e) acquire the measurement m_t at new position.
   (f) with the a priori pdf $p(m_t|s_t)$ modify the belief state at time t: $Bel_t(s^k) = p(s^k_t|H_t) = c_t p(m_t|s_t)\widehat{Bel}_t(s)$, where c_t is the current normalization coefficient (the sum of belief state distribution should be equal to 1).

2.3 Illustration of a 2-dimensional self-localization

In our experiments only two degrees of freedom of the camera were allowed: a translation along the X and Y axes by unit steps. Hence, the possible actions were: $\{a_t = (dx, dy)|dx, dy = \{-1, 0, 1\}\}$. A single image corresponds to one particular view of the scene (see Figure (2)). The measurement system in this case is performed by an image analysis system, which detects a measurement vector for each image.

3 Image feature detection methods

We have implemented three types of measurements, which are based on global image features. Although,

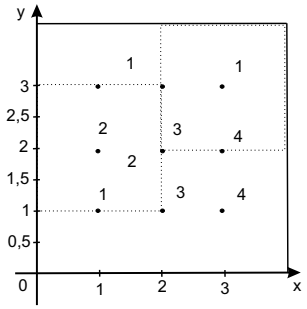


Figure 2: Illustration of the self-localization process. A 2-D scene is split into n x n views, where each position corresponds with a single system state. Due to image analysis the appearance of digits (from 1 to 4) is detected and summarized by a measurement vector.

they are of simple nature and they usually provide not sufficient information to distinguish in one step between all possible states (views), they are very helpful to demonstrate the efficiency of the discrete localization method. The following types of the measurement vector were created:

1. *MeanVar* - the three mean and three standard deviation values of the R, G and B channels of the image.

2. *FFT6* - the modules of first 6 components of a Fourier transform of the image.

3. *Hist6* - the three dominating color components in the image with their density values.

3.1 Learning the a priori pdf

The a priori pdf $p(m|s)$ should be computed during the learning phase. But the number of possible measurement vectors is infinite, usually there are continuous-valued components of m. In practice this pdf can be made explicit only during the active work. In the learning phase we compute and store the feature vectors associated with each discrete state.

During the active work the feature vector of current view is detected (assuming a previous normalization of the scene illumination or camera contrast).

The a priori pdf $p(m_{k+1}|s)$ is implicitly defined, as we can compute for each state s^k the value of a Gaussian distributed pdf, with mid point equal to zero, for the the distance of $w|m_{k+1} - m(s^k)|^2$ (where w is a weighting vector that adjusts the intervals of particular components to some common interval).

3.2 Global mean and standard deviation

In the first measurement method the vector m consists of global features of given view. In the learning phase for each view the mean values and standard deviation values of each color component are computed.

3.3 Histogram-based features

Image features that are computed from the histogram of the image are usually of global character. At first for each color component its histogram is obtained, i.e., the density distributions of possible intensity values. We have studied an extensive case, where in addition to the total number of pixels having given intensity, the position of image regions with given intensity is detected (represented by its mass center and boundary box). This approach resulted in a matrix of measurements, rather than a vector. In experiments, even a simplified approach, that uses a 6-elementary feature vector only, turned out to be sufficiently robust. The feature vector consisted of three pairs of values $(I_k, \rho(I_k)), k = 1, 2, 3$ (I_k - the k-highest intensity value, $\rho(I_k)$ - the number of pixels with intensity I_k).

3.4 Frequency-based features

For a square image of size NN, the two-dimensional FFT is given by:

$$F_{(k,l)} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j) e^{-i2\pi(ki/N + lj/N)}, \qquad (7)$$

where $I(i,j)$ is the image in the spatial domain and the exponential term is the basis function corresponding to each point $F_{(k,l)}$ in the Fourier space. The equation (7) can be interpreted as: the value of each point $F_{(k,l)}$ is obtained by multiplying the spatial image with the corresponding base function and summing the result. The basis functions are sine and cosine waves with increasing frequencies, i.e. $F_{(0,0)}$ represents the DC-component of the image which corresponds to the average brightness and $F_{(N-1,N-1)}$ represents the highest frequency.

3.5 Test results

Several scenes with different illumination conditions were available for testing (Figure 3). Three alternative measurement vectors were tested:

1. *MeanVar*: $m = [m_1, m_2, m_3, \sigma_1, \sigma_2, \sigma_3]$

Scene	Type	MeanVar	FFT6	Hist6
Lift	mean	0.28396	0.54611	0.36141
Map	mean	0.42009	0.71208	0.88932
All	mean	0.48502	0.39076	0.53752
Lab	mean	0.39041	0.39442	0.48202

Table 1: The means of 'p-values' between pairs of feature vectors, learned for the original scene.

2. *FFT6*: $m = [F_{(0,0)}, F_{(0,1)}, F_{(0,2)}, F_{(0,3)}, F_{(0,4)}, F_{(0,5)}]$
3. *Hist6*: $m = [I_1, I_2, I_3, \rho(I_1), \rho(I_2), \rho(I_3)]$

As indicated on fig. 4 the intensity and contrast of current scene is significantly changed if compared to the original scene, on which the system has learned. Also the positions of views are shifted by few pixels from their original positions. With a particular system state only a small view (via an image of size 256x256) of the scene is available. The goal state (on the right), the current (unknown to the system) view (in the center) and the default initial best state (on the left) of the real scene.

3.6 Statistics of the feature vectors

Let us first verify the statistical correctness of the proposed image feature vectors.

The *p-value* of two distributions expresses the correctness of a hypothesis, that both distributions are statistically equivalent. If the p-value is equal to zero, then the above hypothesis is wrong and both features can be treated as being different. This definition can be extended to more than two distributions.

We have examined the behavior of our three sets of image features, when the image analysis was applied to our four 2-D scenes. Table 1 summarizes the results of 'p-val' computed for all pairs of the measurement vector (pairs of states of the original scene). The values of 'p-val' near zero indicate, that the features are distinct. From this point of view the feature set *MeanVar* is more distinctive, than our *FFT6*-set (with the third set *Hist6* located in-between). Table 2 summarizes the values of *p-value* computed for all pairs of feature vectors, where the first element of the pair corresponds to the state of the original scene, and the second element - to the compatible state in the real scene. Remember, that the compatible views are shifted one-against-the-other by few pixels, and the intensity and contrast of the camera have also changed.

Figure 3: Examples of 2-D scenes that were used for testing: *All, Lab, Lift, Map.*

The *optimal* feature set achieves a p-value of nearly 1, what means, that the features for learned view and real view are the same. Form this point of view this table documents, that the feature set FFT6 is performing best of all, i.e., the other two sets are very sensitive to changes of lighting and camera positions.

Figure 4: The scene used for learning (top image) differs from the real scene (bottom image) by illumination conditions and a slight shift of camera positions.

3.7 The quality of self-localization

For each 2-D test scene (in total - four scenes were used) and for each measurement method we have run the self-localization process 100 times, with randomly chosen start and goal states. A particular self-localization process is illustrated on figure 5. At the start point the belief state distribution is an uniform distribution. After 2-4 steps the appropriate state that corresponds to the real position can already be selected - the belief state value for such state dominates already the beliefs of remaining states.

In table (3) we provide data illustrating the correctness (quality) of self-localization tests in different scenes and measurement methods. The *FFT6*-set allowed for error-free self-localization, the *MeanVar*-set performed very well, whereas the *Hist6* was very sensitive to lighting and position changes.

Scene	Type	MeanVar	FFT6	Hist6
Lift	min	0.2345	0.7452	0.1920
	max	0.9985	0.9999	0.9872
	mean	0.8833	0.9606	0.7682
Map	min	0.3015	0.9322	0.2643
	max	0.9999	0.9999	0.9943
	mean	0.9429	0.9856	0.8953
All	min	0.1733	0.1738	0.1695
	max	0.9998	0.9997	0.9965
	mean	0.8866	0.9456	0.8240
Lab	min	0.5867	0.5613	0.5230
	max	0.9976	0.9966	0.9985
	mean	0.9179	0.9158	0.8896

Table 2: The minimum, maximum and mean values of the 'p-value' for pairs of measurement vectors for compatible states in the original and real scene.

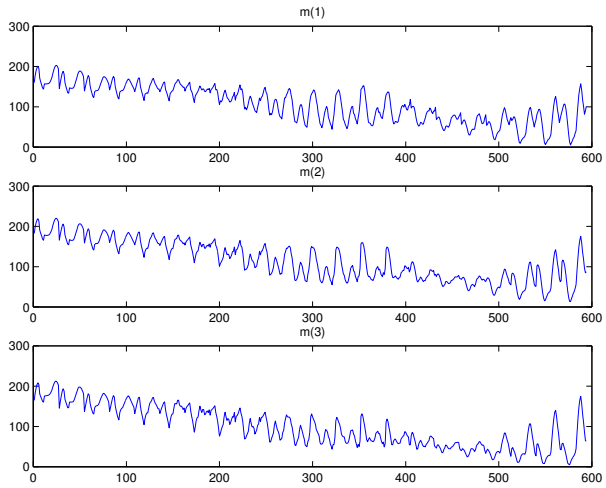


Figure 5: The distribution of the first 3 measurement features of the *MeanVar*-set corresponding to states.

4 Summary and Conclusions

A self-localization method for (partly) known indoor environment was designed and it was experimentally proved to be robust and effective. Three different environment measurement algorithms, based on image analysis, were proposed and tested. It was shown that even for natural scenes, the use of even a small set of image features, expressing only global information of a particular view, is sufficiently robust. An obvious precondition for using global image features is the scene normalization, i.e., the images of the scene (environment) acquired during the active navigation should

Scene state num.	MeanVar	FFT6	Hist6
Lift-352	89	100	54
Map-234	99	100	78
All-594	99	100	71
Lab-150	87	100	51

Table 3: The number of successful runs of the self-localization process (for 100 tests in total).

be non-linearly scaled to adapt to the contrast and intensity of the scene in the learning phase. Otherwise the detection of discrete image features is preferred, also this is out of scope of this paper. The global image features are sufficiently robust to overcome small geometric perturbations in image acquisition, i.e., a shift of the view by several pixels with respect to the learned position.

Acknowledgments

This work was supported by *Research Center for Control and Information-Decision Technology (CATID)* at Warsaw University of Technology, Poland.

References

[1] J. Borenstein, H. Everett, L. Feng: *Navigating Mobile Robots*. Wesley, Mass., 1996.

[2] W. Burgard, A. Cremers, D. Fox, D. Hahnel, G.Lakemeyer, D. Schulz, W. Steiner, S. Thrun: Experiences with an Interactive Museum Tour-Guide Robot. *Artificial Intelligence*, vol. 114 (1999), No. 1-2, 3-55.

[3] Denzler J., Zobel M.: Automatische farbbasierte Extraktion naturlicher Landmarken und 3D-Positionsbestimmung. V. Rehrmann (ed.): *Vierter Workshop Farbbildverarbeitung*, Fohringer-Vg., Koblenz, 1998, 57-62.

[4] D. Fox, W. Burgard, S. Thrun: Markov Localization for Mobile Robots in Dynamic Environments, *Journal of Artificial Intelligence Research*, vol. 11(1999), 391-427.

[5] Jochem T.M.: *Vision Based Tactical Driving*, Carnegie Mellon University, Ph.D. diss., CMU-RI-TR-96-14, 1996.

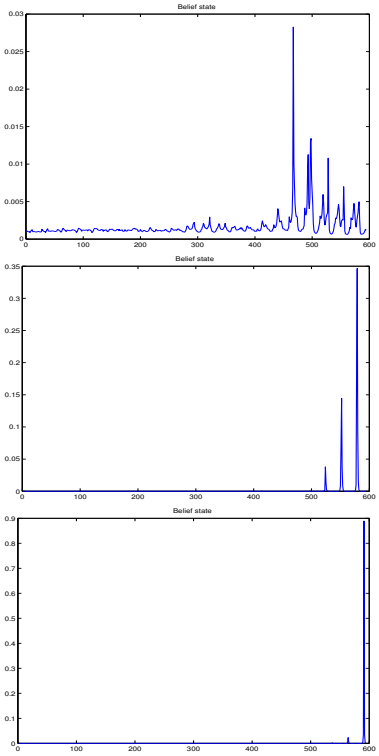
[6] Kasprzak W.: Adaptive computation methods in image sequence analysis. *Prace Naukowe - Elektronika*, No. 127 (2000), Warsaw Univ. of Technology Press.

[7] Kasprzak W.: Analiza obrazów cyfrowych w zastosowaniu do dyskretnej samo-lokalizacji. *Pomiary-Automatyka-Robotyka*, vol. 5 (2001), No. 4, PIAP, Warsaw, 5–10.

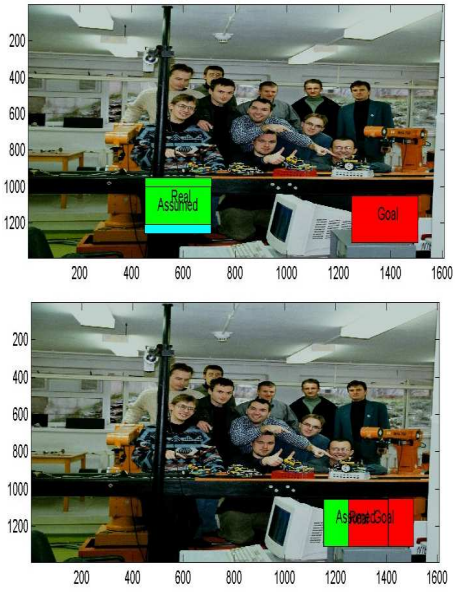
[8] Thorpe C. (ed.): *Vision and Navigation: The Carnegie Mellon Navlab*. Kluwer Academic Publ., Norwell, Mass., 1990, 25-38.

[9] Masaki I.: *Vision-based Vehicle Guidance*. Springer, New York, Berlin-Heidelberg etc. 1992.

[10] Maurer M., Behringer R., Thomanek F., Dickmanns E.D.: A compact vision system for road vehicle guidance. *13th Int. IAPR Conference on Pattern Recognition*, Vienna, Aug. 1996, 313–317.



(a)



(b)

Figure 6: Illustration of the self-localization process: (a) the second (after the initial uniform distribution) 6th and 16th belief state distribution; (b) the assumed, real and goal state are shown after 5 and 16 steps - approaching the goal state.