INŽENÝRSKÁ MECHANIKA 2005

NÁRODNÍ KONFERENCE

s mezinárodní účastí

Svratka, Česká republika, 9. - 12. května 2005

MARKOV LOCALIZATION FOR MOBILE ROBOTS: SIMULATION AND EXPERIMENT

J. Krejsa[*], S. Věchet[+]

Summary: *Localization of the robot is a task of estimating robot position in known environment from sensor observation. The paper describes basic principles of Markov localization technique, succesfully used for localization task. Method is robust against sensor errors and can deal with global uncertainty when robot position is completely unknown. Both simulation and experimental verification of method usability are included.*

1. Introduction

The problem of robot localization is usually defined as estimating robot position within known environment (a model of environment must be given in some form, e.g. topological map, grid based description of obstacles, etc.) based on observations, usually odometric readings, proximity sensors, cameras, etc. There is a number of algorithms to deal with this problem, most of them falls into local localization category, when initial location of a robot is known and the algorithm compensates odometric readings errors based on sensor observation. A new localization algorithm called Markov localization which based on probabilistic approach was proposed recently (Fox et. all, 1999; Bougard et. al, 1998). This algorithm is capable of dealing with global uncertainty when robot location is completely unknown.

2. Markov localization basics

The key idea of Markov localization is to calculate the probability of “being there” for all possible locations in environment. Such belief of being at certain position l in certain time is denoted as $Bel(L_T = l)$. Position l is robot position in x, y, α space, where x, y are cartesian coordinates of the robot and α is it’s orientation. When initial position of the robot is unknown, $Bel(L_0)$ is uniformly distributed. Probability distribution is updated whenever robot obtains data from sensors, which in general can be divided into two groups: odometry

[*] Ing. Jiří Krejsa, PhD., ÚT AV ČR, pobočka Brno, Technická 2, 616 69, Brno, Czech Republic; tel: +420 541142885, email: jkrejsa@umt.fme.vutbr.cz

[+] Ing. Stanislav Věchet, PhD., VUT Brno, Technická 2, 616 69, Brno, Czech Republic, email: vechet@fme.vutbr.cz

readings (robot moves), environment sensors (usually proximity sensor of some kind – infrared, ultrasonic, laser).

When robot moves and odometry reading is obtained, the belief is updated as $Bel(L_T = l) = \sum_{l'} p_a(l \mid l') Bel(L_{T-1} = l')$, where $p_a(l \mid l')$ is the probability that robot is moved from location l to new location l' with the action a. One can assume that in translation and rotation the errors are normally distributed and therefore two independent zero-centered gaussian distributions can be used to model the movement, when variances of distributions are proportionally dependent on the length of measured movement.

When an environment sensor reading s_T is obtained, the belief is updated as $Bel(L_T = l) = \alpha_T p(s_T \mid l) Bel(L_{T-1} = l)$, where $p(s_T \mid l)$ is probability of sensing s_T at l (it specifies the probability of observations in different locations of environment) and α_T is normalizer that keeps sum of $Bel(l)$ for all l equal to 1.

Furthermore the representation of robot's belief $Bel(l)$ and conditional probability $p(s_T \mid l)$ computation must be determined.

There are various ways how robot belief can be represented, e.g. Kalman filtering based approach (see e.g. Smith, 1990), topological approach (see e.g. Kaelbling, 1996), etc. Those techniques are unfortunately limited in some ways (Kalman filters require to know the starting location of the robot, topological approach gives only rough sense of location). To overcome those limitations, Bougard et al. (1998) suggests using fine-grained, regularly spaced grid, with spatial and angular resolutions depending on sensors types used and also on state space size, typically the spatial resolution is between 15-30 cm, angular resolution between 2-5 degrees.

Only proximity sensors are used as environment sensors furthermore. As a perception model of such sensor Fox et al. (1999) developed a sensor model calculating $p(s_T \mid l)$ based only on the distance o_l to the closest obstacle in the map along the direction of the sensor: $p(s_T \mid l) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(s_T - o_l)^2}{2\sigma^2}}$, where standard deviation σ of the distribution depends accuracy of the world model and accuracy of the sensor.

Figure 1. gives example of such distribution for infra-red sensor. The distance to the closest obstacle is 270 mm. The figure also shows measured probabilities of measurements obtained with infrared sensor Sharp GP2D12.

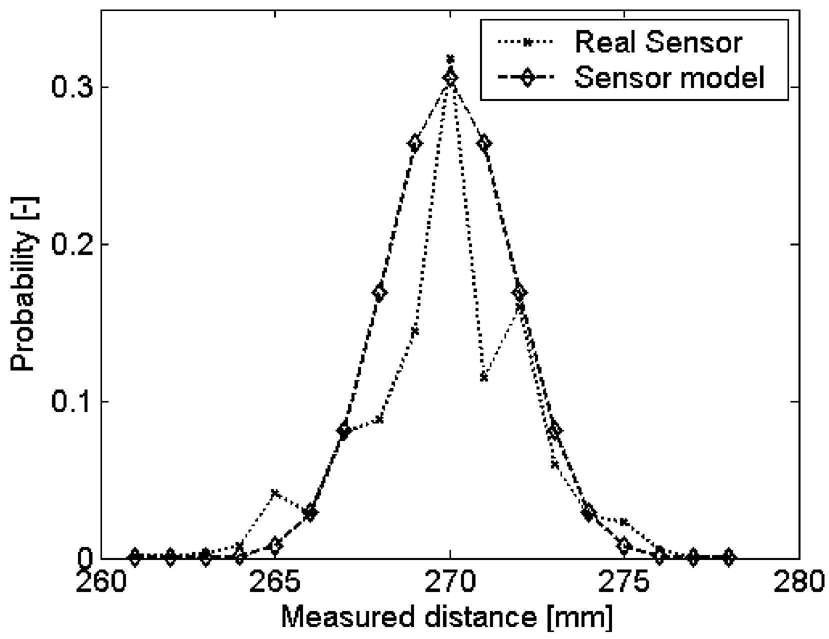

Figure 1. Proximity sensor perception model vs. measured values (Sharp GP2D12 IR sensor)

3. Implementation issues

When using fine grids as state space representation, it is obvious that the memory and mainly computation requirements can grow substantially. For example environment of $10 \times 10m^2$ with angular grid resolution of 2° and a cell size of $10 \times 10cm$ brings state space consisting of 1 800 000 states. Such matrices can not be updated in real time given current computer speed (updates are required for each sensory input and step movement of a robot). Therefore some techniques are required to speed up the updates for the method to be usable in real world applications.

First useful technique is pre-computation of sensor model. Perception model $p(s_T | l)$ depends only on distance o_l to the closest obstacle in the map along the direction of the sensor. Therefore it is possible to pre-compute and store those distances for all possible robot locations.

Second technique is using a selective update. During update density quickly concentrates on the grid cell which represents the real location of the robot (and possibly its closest neighbors). Therefore it is reasonable to approximate $p(s_T | l)$ for those cells, whose belief is below certain threshold by averaging over all possible locations of the robot $p(s_t) = \sum_l p(s_t | l) p(l)$.

4. Simulation tests

Markov localization as described above was implemented using Borland Delphi 7 and first tested on simulated data. Test world was described with 20 cm resolution on 50×50 grid and angular resolution of 5° (thus forming 180 000 states). Sensor readings were weighted with 20% noise. Simple filter for IR-sensor data was applied (in computer simulation the filter only erased some of the “readings” making the Markov localization job more difficult). An example of the belief distribution development for map with three obstacles is shown on Figure 2. Belief distribution is projected to 2D (darker the color higher the accumulated belief over all orientations) with sensor readings and robot trajectory included. All frames are independently normalized.

First picture nicely illustrates the principle of Markov localization – only a single reading is obtained (second obstacle is behind the first one, third obstacle reading was eliminated by the filter), therefore algorithm adds “circles” of measured distance radius centered around obstacles. Second frame shows updated belief after a movement reading is obtained. All the frames come in sequence: (environment reading, translation, environment reading, rotation). One can see that after a few updates a standalone peek is created, corresponding to the real position of the robot.

5. Simple real robot tests

Computer simulation results encouraged further testing on real robot. A simple robot was built as a testing platform, equipped with motors, sensors and basic electronics. Table 1. shows robot basic technical parameters. This simple robot is described in detail in Věchet, Krejsa, 2005.

Table 1. Test robot parameters.

Parametr	**Value**
Engine type	Two stepper motors TEAC
Sensor equipment	Two GP2D12 sensors
Maximal speed	10cm/s
Onboard electronics based on	ATMEGA8
Power supply	15V,0.3mA

Test results shown in Figure 3 were obtained with Markov localization state space grid of the same resolution as used for simulations. Belief distributions are again projected to 2D, snapshots sequence is also the same as in Figure 2.

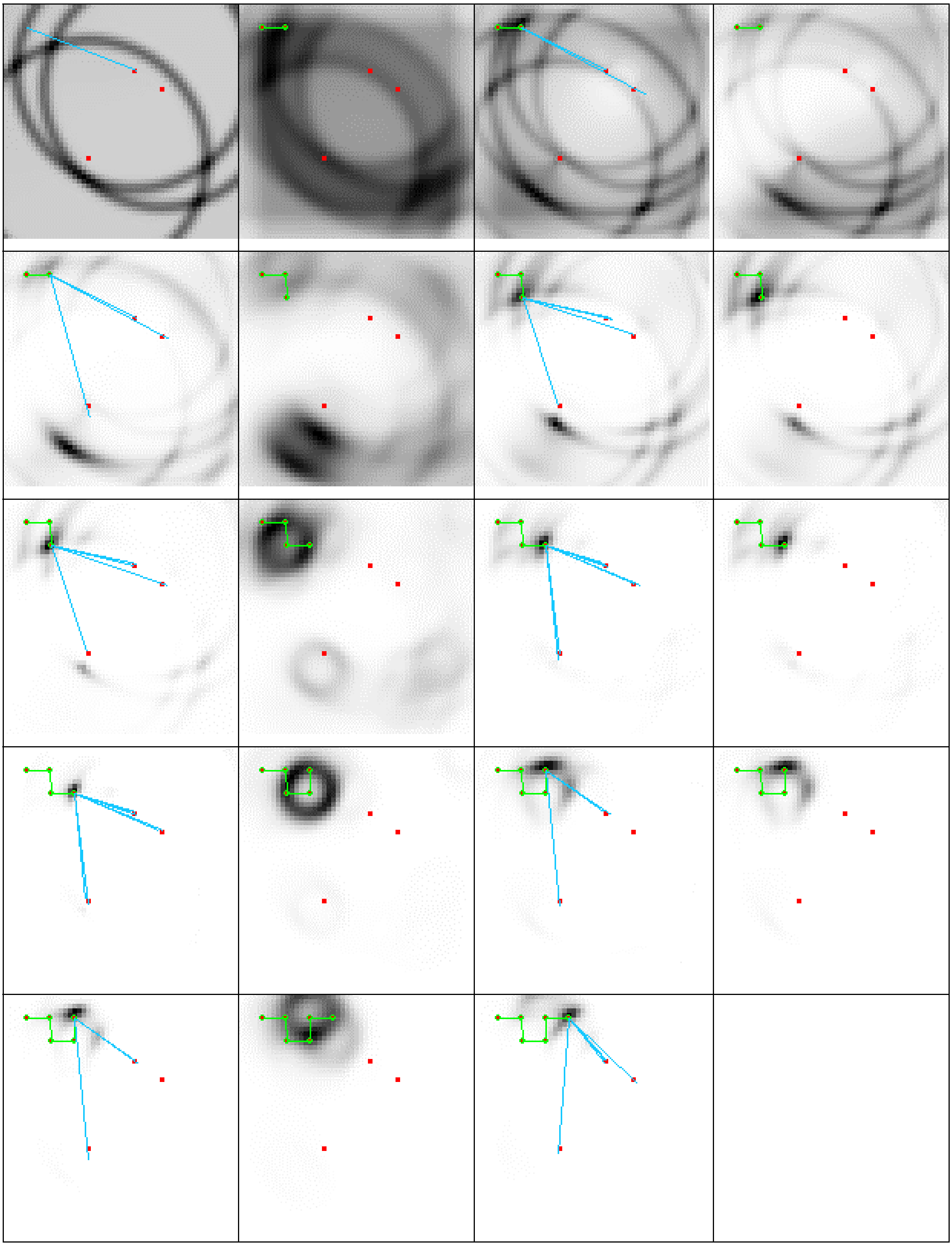

Figure 2. Development of belief distribution during localization – simulation

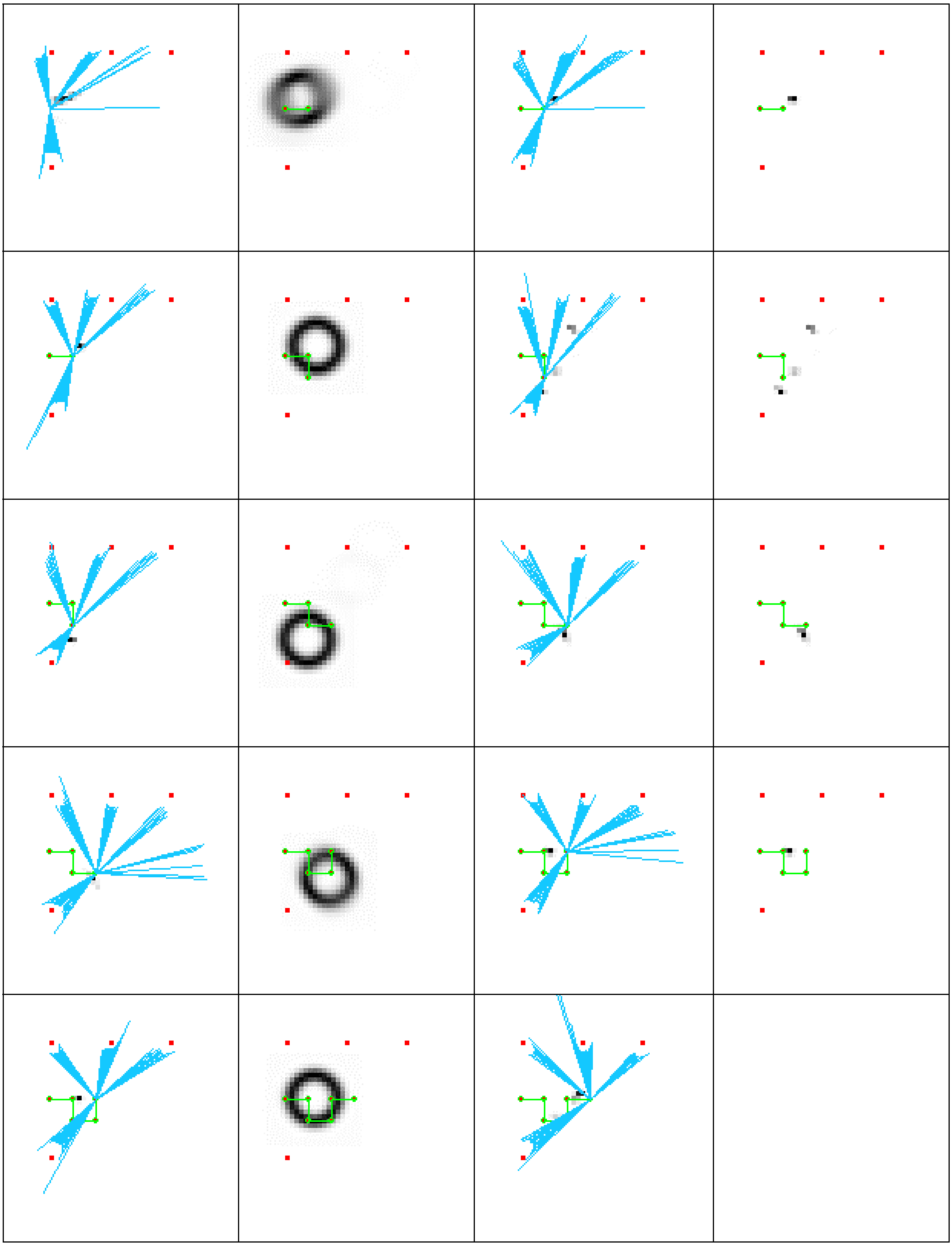

Figure 3. Development of belief distribution during localization – real robot

6. Conclusions

Markov localization is a robust technique which is capable of correct estimation of robot position even under global uncertainty (robot position is completely unknown). The key idea of the method is to calculate the probability distribution over all possible locations in state space. The distribution is updated whenever robot receives sensor inputs – odometry readings or environment sensor readings.

Markov localization is capable of dealing with sensor noise up to high levels, therefore cheap infrared or ultrasound sensor can be used and it is also robust to approximate models of the environment – occupancy grid maps can be succesfully used.

The method is computationally greedy and requires special techniques to reduce computational requirements, however those increase memory requirements. Computational requirements are connected to the method only limitation which is the state space size. However for indoor applications of reasonable magnitude is the cost of algorithm reasonable.

7. Acknowledgement

This work was supported by Czech Ministry of Education under project MSM 0021630518 "Simulation modelling of mechatronics systems".

8. References

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

Smith R., Self M., Cheeseman P.: Estimating Uncertain Spatial Relationships in Robotics, in I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, Springer Verlag, 1990

Kaelbling L., Cassandra A. Kurien J.: Acting Under Uncertainty: Discrete Bayesian Models for Mobile-robot Navigation, in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1996

Bougard W., Derr A., Fox D., Cremers A.: Integrating Global Position Estimation and Position Tracking for Mobile Robots: the Dynamic Markov Localization Approach, in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998

Věchet S., Krejsa J.: How to build a robot with no money, merkur, lego and old stepper motors. in *Proc. of Engineering Mechanics 2005*, Svratka