Towards Outdoor Localization Using GIS, Vision System andStochastic Error Propagation

Mikael Kais, Stéphane Dauvillier, Arnaud de la Fortelle, Ichiro Masaki, Christian Laugier

2nd International Conference on Autonomous Robots and Agents
December 13-15, 2004 Palmerston North, New Zealand

Towards Outdoor Localization Using GIS, Vision System and Stochastic Error Propagation

Mikael Kais[ab], Stéphane Dauvillier[a], Arnaud De La Fortelle[a], Ichiro Masaki[d], Christian Laugier[c]

INRIA, IMARA group[a]
ENSMP, CAOR group[b]
INRIA, E-MOTION group[c]
MIT, MTL[d]

Abstract

Localization with respect to a reference model is a key feature for mobile robots. Urban environment offers numerous landmarks that can be used for the localization process. This paper deals with the use of an environment model stored in a Geographic Information System, to drive a vision system i.e. highlights what to look for ? and where to look for ? This task is achieved by propagating uncertainties along the image acquisition system.

Keywords: Localization, GIS, vision, GPS, error

1 Introduction

Localization of a robot refers to the task of finding the position and attitude of the robot with respect to a model. Two complementary methods are usually used for this task, methods relying on the knowledge of an initial position and the integration of motion of the vehicle and those based on position and attitude measurement with respect to a reference model. Indeed the first category have systematic and random error, so that time integration leads to divergence. That is why it is necessary to regularly refine the localization of the robot by performing position and attitude measurements with respect to a model to keep the localization accurate. The full process can be broken down in four steps :

- Extraction of Landmarks from the sensory data;
- Matching of sensory landmarks with model landmarks;
- Computation of the transformation with respect to the model;
- Position and attitude refinement of the robot.

1.1 Motivation

Numerous types of landmarks are used in mobile robotics for localization purpose. Some methods are based on satellites like GPS positioning, others use reflective poles detected by a laser-scanner, magnets or transponder buried in the pavement offer an other alternative. In the case of urban environment GPS can be affected by some mask due to some occlusion by buildings between the mobile robot and the satellites. An alternative to the equipment of a city with some markers is the use of the environment itself. Indeed urban environment offers numerous primitives like buildings, trees, traffic signs, road boundaries that can be used for the localization process. In this paper we present how an environment model integrated in a GIS (Geographic Information System) can be advantageously used to drive a vision system i.e. highlight some Regions Of Interest in the image where the vision system can trigger some specialized primitive detectors.

1.2 Prior work

In the last decades numerous authors have focused their interests towards the localization of robots. Some of them perform indoor localization, in [3], authors use a stereo head to detect vertical edges to perform 3D reconstruction as well as localization. In [14], annotated map concept is presented, tasks related to spatial locations (e.g. lane following, intersection recognition...) are triggered from the database. In [13], authors uses Digital Elevation Map to localize a robot in a mountain environment. In [4] and [6], authors present a one decade work on active vision for car like robot.

One of the key feature is the matching of landmark from the sensory data with the model, in [7], the authors build an environment model from car navigation type map and use a simple distance in the image to match both primitives, poor robustness is inherent to the method. When geometry is available [10] geometric invariant properties can be used to match laser-scanner impacts with a geometric model representing position of reflective poles. Perceptive field or appearance model offer invariant properties that can be advantageously used [11].

A alternative way to simplify the matching procedure is to use specialized operator like white lane marker detector. In [2] a front camera is used to recover the lane marking that is matched with a precise map. In [8], lateral cameras are used to estimate distance and orientation of the lane marking, matching with an accurate map is performed to refine the localization of the vehicle.

This paper focus on the use of a known model to highlight some Region Of Interest, where primitives of given classes have a given probability to be.

1.3 Problem description

At time t, the environment E is imaged by a sensor as e_0 using a multivariate function f_{p_0}: $e_0 = f_{p_0}(E)$. The parameter p_0 represents all the variables describing the state of the sensor: position, attitude, intrinsic parameters of the sensor, etc. At the same time, a model of the environment M is projected in the sensor space as m_0: $m_0 = f_{p_0}(M)$. We assume the parameters are known at time t (hence this is the same function f_{p_0}) and, since the model M fits the environment E, both images e_0 and m_0 fit.

At time $t+\Delta t$, the integration of the motion gives a new set of parameters p' when the real parameters are p, so we get, since the environment has not changed:

$$e = f_p(E) \qquad \text{and} \qquad m = f_{p'}(M).$$

But e and m does not fit anymore. The goal of our method is to calculate the real parameters p (including the localization of the mobile) by fitting the images e and m. We do this by letting p' vary, i.e. in a sense we solve in p' the equation $f_{p'}(M) = e$. Once this is done, we conclude that $p' = p$ so the localization is achieved.

Now, p' and p are very close so that we can reduce the search for environment primitives inside the image e. Moreover, most of the time, this method allows to track the primitives: once a pattern has been localized in the image e, it can be context-free set as only one object in the model. This reduces significantly the computation time.

The region of search in the image e for an object of the model is calculated by estimating the error for each parameter of p'. More precisely, we estimate the two first moments for each parameter and propagate the uncertainty of points in the environment E to the image e. Therefore points are transformed to random variables which two first moments are estimated. Hence the points are to be searched in ellipses, which the experiments prove to be indeed small. The GIS is then used to activate an image processing operator to look for a specific primitive inside a small area delimited by these ellipses.

2 Modeling

In this section, we focus on modeling odometry integration and projection of a 3D point M of the environment model into a 2D point m in the image plane of a camera (see Fig. 2).

Position and attitude, also called configuration of the vehicle wrt. a reference frame R_0 is estimated by knowing an initial configuration and time integration of small displacements of the vehicle. Measurements recovered from the vehicle are the front wheel steering angle and the linear speed.

Using the known configuration, the speed and the front wheel steering angle of the vehicle, recursive estimation of the configuration of the robot can be done in two steps:

- Using the known configuration, compute the linear and the angular velocities of the vehicle with respect to the reference frame R_0;
- Compute the new configuration of the robot using the previous configuration and time integration of linear and angular velocities of the vehicle.

Projection of a $3D$ point in the image can be done in three steps:

- A rigid transformation is applied to the 3D points defined in the reference frame R_0 (or world frame) to get point coordinates in the camera frame R_c. Coordinates of M in R_0 and R_c are respectively denoted by M_0 and M_c;
- 3D points in the frame R_c are projected into 2D points in the normalized camera frame R_n, i.e. a camera with a focal length of one. Coordinates of m in R_n is denoted by $m_n = (x_n, y_n)$;

- 2D points in R_n are scaled with the appropriate focal length and translated to get points in image top-left coordinate system R_i, where the coordinates of m are denoted by $m_i = (u, v)$.

2.1 Odometry

In this part we focus on time integration of small displacements. Simple bicycle kinematic model Fig.1 is considered to model the car kinematic. Assuming the flatness of the environment where the car is running, position and attitude of the vehicle are resumed to the position of the vehicle in the 2D plane (Oxy) defined by $x(t)$ and $y(t)$ and orientation of the car with respect to the (Ox) axis given by $\theta(t)$. Measurements from the vehicle are the speed $V(t)$ of the middle point of the rear axis and the steering angle $\alpha(t)$.

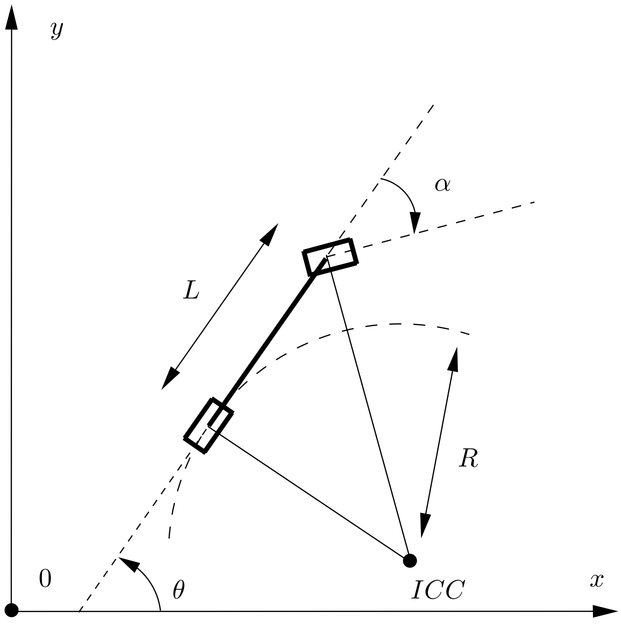

Figure 1: Bicycle model

Relation between linear speed V and angular speed ω is given by: $V = \omega R$, where R stands for the radius of curvature of the rear wheel i.e. the distance between the middle of the rear axis and the Instantaneous Center of Curvature. Geometric considerations on Fig. (1) yields $R\tan(\alpha) = L$. Combining both relations gives the expression of the angular velocity of the vehicle:

$$\omega = \frac{V}{L}\tan(\alpha) \tag{1}$$

Linear and angular velocity of the vehicle in the reference frame R_0 are given by :

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} V(t)\cos(\theta(t)) \\ V(t)\sin(\theta(t)) \\ \omega(t) \end{bmatrix} \tag{2}$$

Position and heading angle of the vehicle in time are given by integrating linear and angular velocities. A discrete integration of (2). gives:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k} + \Delta t V_k \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \\ \frac{1}{L}\tan(\alpha_k) \end{bmatrix} + \eta_k \tag{3}$$

where position and heading angle at instant $k+1$ (denoted by X_{k+1}) is given by the position and heading angle at instant k (denoted by X_k) plus a small displacement ΔX_k and some noise η_k coming from the discretization of the system, so that the dynamic recurrence (3) is simply written

$$X_{k+1} = X_k + \Delta X_k + \eta_k. \tag{4}$$

2.2 Rigid transformation

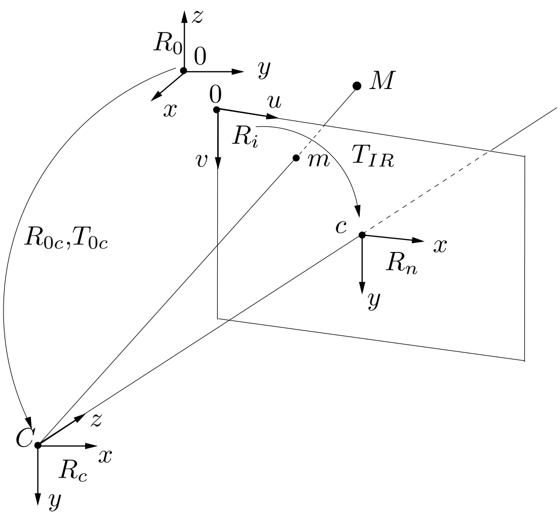

Figure 2: Pinhole camera model

The change of coordinates between R_0 and R_c is modeled by a rigid displacement

$$M_0 = R_{0c}M_c + T_{0c} \tag{5}$$

where R_{0c} and $T_{0c} = [t_x, t_y, t_z]^T$ represent respectively the rotation and the translation between the reference frame and the camera frame. Euler's angles are used to represent rotations, hence the rotation matrix R_{0c} is the product of three $3 * 3$ matrices that represent a rotation on each axis:

$$R_{0c} = R_z(\theta)R_y(\phi)R_x(\psi) \tag{6}$$

with

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$R_y(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \quad (8)$$

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \quad (9)$$

Therefore the coordinates of M in the camera frame is given by:

$$M_c = R_{0c}^{-1}(M_0 - T_{0c}) \quad (10)$$

Rotation matrices are orthogonal so that ($R^{-1} = R^T$) and

$$M_c = R_x^T R_y^T R_z^T (M_0 - T_{0c}) \quad (11)$$

2.3 Perspective projection

The Pinhole model is assumed for the camera (see for instance [5]): m represents the intersection between the optical ray (CM) and the retinal plane. In R_n the camera is normalized, i.e. the focal length Cc equals one and $m_n = (x_n, y_n)$ is obtained from $M_c = (x_c, y_c, z_c)$ by a perspective projection. Thales' theorem yields:

$$\begin{cases} x_n = \dfrac{x_c}{z_c} \\ y_n = \dfrac{y_c}{z_c} \end{cases} \quad (12)$$

2.4 Scaling and translation

Transformation from normalized frame R_n to the image frame R_i is done by scaling and translation:

$$\begin{cases} u = f_u x_n + u_0 \\ v = f_v y_n + v_0 \end{cases} \quad (13)$$

where f_u and f_v are the horizontal and vertical focal lengths of the camera in pixels unit, u_0 and v_0 are the coordinates of the principal point c in the image frame R_i also in pixel.

3 Uncertainties propagation and Kalman filtering

3.1 Odometry and DGPS Kalman filtering

The system can be modeled by the following equation

$$X_{k+1} = g(X_k, U_k, \eta_k)$$

Where g is the evolution function of the system, $U_k = [V_k \ \alpha_k]^T$ is the command vector with V and α respectively the speed and the front wheel steering angle of the car. Estimation of the configuration of the vehicle (hence the camera) is performed by fusing car odometry with a centimetric Differential GPS. Extended Kalman filtering is performed using the two known steps i.e. prediction and update [15].

When no measurement is available, the best estimation of the state is given by integrating odometry. Associated covariance matrix Γ_{k+1} is computed by propagating covariance of the last state and the command vector. Jacobian matrices A_k and B_k are respectively the partial derivate of the evolution function g wrt. the state vector X_k or the command vector U_k.

$$\begin{aligned} \overline{X}_{k+1} &= g(\hat{X}_k, \overline{U}_k, 0) \\ A_k &= \tfrac{\partial g}{\partial X_k} \\ B_k &= \tfrac{\partial g}{\partial U_k} \\ \Gamma_{X_{k+1}} &= A_k \Gamma_{X_k} A_k^T + B_k \Gamma_{U_k} B_k^T + \Gamma_{\eta_k} \end{aligned}$$

When a measurement $Y_{k+1} = h(X_{k+1}, \nu_{k+1})$ from the GPS is available, the above estimation can be corrected. Where h is the measurement function and ν_{k+1} some noise. The kalman gain K can be computed using the Jacobian matrix of h wrt. to the state C_{k+1}. The best estimate and associated covariance matrix can be deduced as follow :

$$\begin{aligned} C_{k+1} &= \tfrac{\partial h}{\partial X_{k+1}} \\ K_{k+1} &= \Gamma_{X_{k+1}} C_{k+1}^T (C_{k+1} \Gamma_{X_{k+1}} C_{k+1}^T + \Gamma_{y_{k+1}})^{-1} \\ \hat{X}_{k+1} &= \overline{X}_{k+1} + K_{k+1}(Y_{k+1} - \overline{Y}_{k+1}) \\ \overline{Y}_{k+1} &= h(\overline{X}_{k+1}, 0) \\ \Gamma_{X_{k+1}} &= (Id - K_{k+1} C_{k+1}) \Gamma_{X_{k+1}} \end{aligned}$$

Where Id stands for the identity matrix.

3.2 Error propagation to the image plane

As seen in the previous section, a projection of a 3D point in the camera frame is a function of 13 variables:

$$m = f(X_c, Y_c, Z_c, \theta, \phi, \psi, t_x, t_y, t_z, f_u, f_v, u_0, v_0) \quad (14)$$

Considering that variables are affected by small perturbations, we linearize the expression using first order Taylor expansion. Each variable is modeled as a random variable characterized by its two first moments i.e. the mean and the variance. The function f can be decomposed as several functions, where mean and variance-covariance matrices are propagated. Reader can refer to [9] for a description of error propagation from one point of the model to the virtual camera image plane.

4 Experiments

4.1 Implementation

The algorithm was implemented on the LARA vehicle of INRIA (Fig. 3). This is a Twingo car type from the Renault car-maker equipped by our laboratory for assisted or autonomous lateral control. There are three onboard controllers, two PCs and one MPC555 microcontroller to perform hard real time control, data acquisition and time-stamping. Both PCs are linked together with an Ethernet link, one hosts a Geographic Information System server, while the other is linked to a camera (Fig. 4) and the MPC555 through a Controller Area Network bus. The MPC555 generates synchronization signal for the camera and recover measurements from the car at a frequency of 20 Hz. Reference position and heading angle of the car (hence the camera) are computed by fusing odometry and measures from an RTK Differential GPS receiver with a Kalman filter.

Figure 3: LARA vehicle

Figure 4: Screen and camera(s)

The GIS server runs PostGRESQL with PostGIS spatial extension for handling Simple Query Language requests. The environment model stored in the GIS is derived from a CAD file made from a geographic survey. Different classes of objects were imported in the database. Algorithm was implemented in C++ and integrated in the RtMaps platform [12]. Given an estimated position of the robot, objects are recovered from the database and displayed in a map like window (Fig. 5). Reader can observe objects of different classes like the road related objects (boundaries, pedestrian crossing), the lamp mast, the trees and building edges.

Objects are also projected in the image plane of a virtual camera then superimposed with the image from the real camera (Fig. 6).

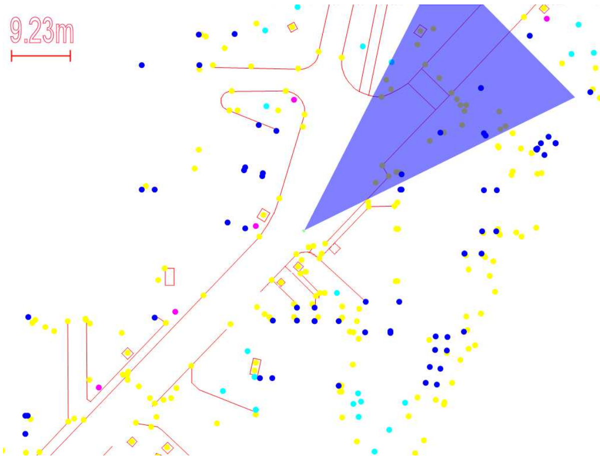

Figure 5: Map

4.2 Uncertainties propagation

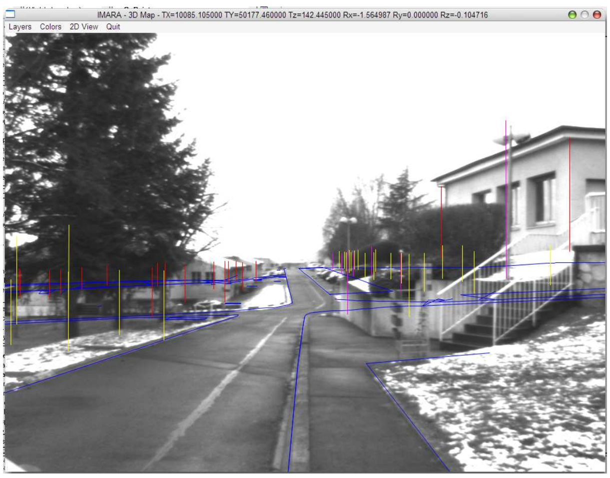

Figure 6: Image model matched

Intrinsic parameters and associated uncertainties of the virtual camera are set to be those recovered from the real camera with a chess board using a Matlab based calibration method [1]. During the experiment, the vehicle was manually driven, estimated position and heading angle were computed with the Kalman filter. Covariance matrix on the state vector were propagated to the virtual camera image plane. Images from the virtual camera was superimposed with images from the real camera. When the real and virtual camera have the same intrinsic and extrinsic parameters, reader observe that both images fit nicely (Fig. 6). Segments of different colors represent different classes of objects :

- Blue for road boundaries and road related objects;
- Red for buildings edges;
- Yellow for trees;
- Pink for lamp mast.

4.3 Experimental procedure

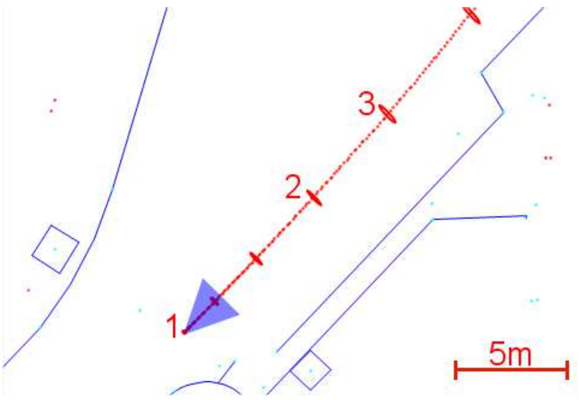

Figure 7: Position of the vehicle

To validate experimentally our approach, the car was driven on a flat surface. Once the Kalman filter has converged (heading angle), state vector and associated covariance matrix of the position and heading angle of the vehicle were associated to position of the virtual camera. In this way uncertainties on the position and heading angle are propagated in the virtual camera image. To validate our approach, the update step with the RTK GPS measure of the kalman filter was not performed so that error continue to grow. Path of the vehicle as well as uncertainties are computed and displayed in a map window (Fig. 7). Uncertainties on position and attitude of the car are propagated in the virtual camera image. Real image and image from the virtual camera including uncertainties are superimposed on Fig. 8 to 10. In this manner, a point of the environment model is imaged as an ellipse. Several ellipses define some Regions Of Interest, where a given primitive e.g. a tree is supposed to be.

4.4 Results

Reader can observe on Fig. 7 the integrated path driven by the car, each position is represented by a red dot. For a better view, uncertainties are given for a couple of positions under the form of an ellipse giving the position of the vehicle with a probability of 0.99. Position 1, 2 and 3 represent respectively the position of the car and estimated uncertainties at time $t_1 = 1.5s$, $t_2 = 5,5s$ and $t_3 = 7s$.

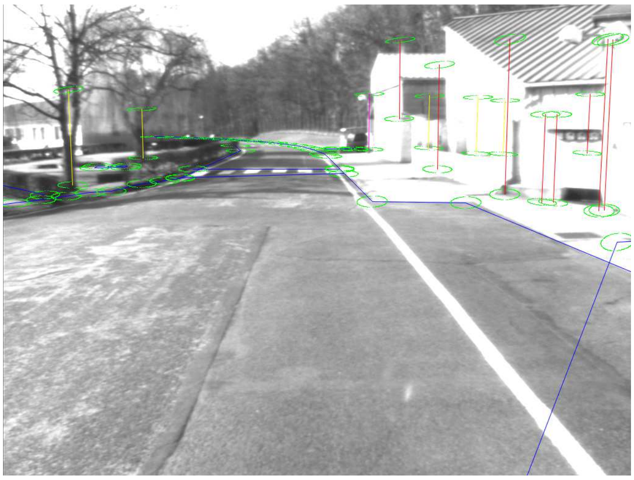

Figure 8: Ellipse drawing at position 1

Reader can observe on Fig. 8 to 10 that the size of ellipses increase, however primitives of the environment stay in the Region Of Interest defined by the ellipses. This result is easily highlighted by looking at the trees on the left, the lamp mast on the right as well as the road boundaries.

5 Conclusion

We have presented in this paper a framework based on a model integrated in a Geographic Information System to drive a vision system using error propagation. Uncertainties on the environment model,

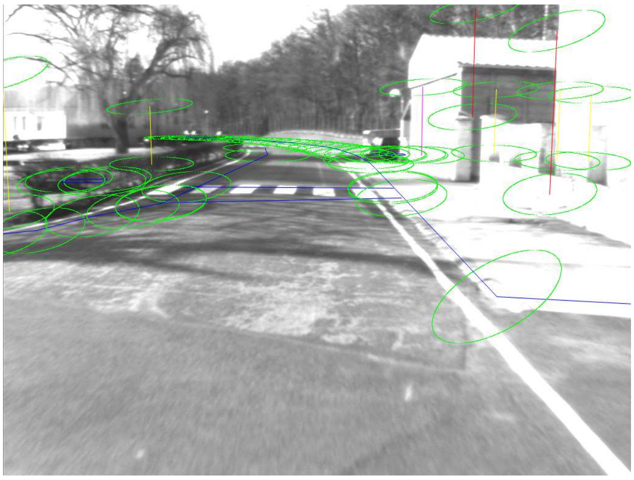

Figure 9: Ellipse drawing at position 2

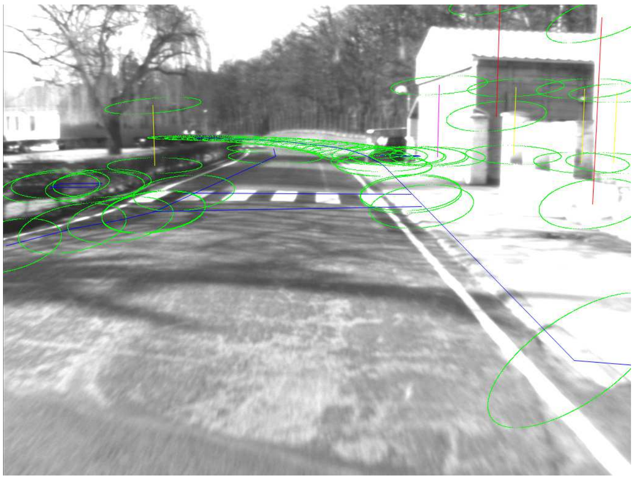

Figure 10: Ellipse drawing at position 3

the position of the vehicle, the attitude of the vehicle as well as on the intrinsic parameters of the camera are propagated to highlight some ellipses in the image plane, each ellipse being the image of a model point. Ellipses define some Region Of Interest where a primitive of a given class (road edges, buildings...) of the model have a known probability to be. If the image sensor is well calibrated (intrinsic parameters), the main sources of uncertainties in the image arise from the position, the attitude of the camera and the environment model. These three parameters are the limitation factors of this method in the sense where the bigger they are, the bigger the ellipses and ROIs will be. Our future work will consist in the use of the model to recover and match model primitives with sensor primitives to refine position and attitude of the vehicle.

6 Acknowledgment

Steeve Morin is greatly acknowledged for its expertise and the programmation of different parts of the system including the Graphical User Interface and the database interface. The authors would like to thank Michel Parent for fruitful discussions. This work would not have been possible without the support of the European Commission Information Society Technology program and the Lounsbery Foundation.

References

[1] J.-Y. Bouguet. Camera calibration toolbox for matlab, 1999.

[2] F. Chausse, V. Voisin, J. Laneurit, and R. Chapuis. Centimetric localization of a vehicle combining vision and low cost gps. MVA, Machine Vision and Applications, 2002.

[3] J. Crowley, P. Bobet, K. Sarachik, S. Mely, and M. Kurek. Mobile robot perception using vertical line stereo. *Robotics and Autonomous Systems, Elsevier Press*, 1991.

[4] E. Dickmanns. A general cognitive system architecture based on dynamic vision for motion control. *Systemics, Cybernetics and Informatics*, 2003.

[5] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[6] R. Gregor, M. Lutzeler, M. Pellkofer, K.-H. Siedersberger, and E. Dickmanns. Ems-vision: a perceptual system for autonomous vehicles. In *IEEE Transactions on Intelligent Transportation Systems*, 2002.

[7] F. Heimes and H. Nagel. Towards active machine-vision-based driver assistance for urban areas. *International Journal of Computer Vision*, 2002.

[8] S. Ieng and D. Gruyer. Merging lateral cameras information with proprioceptive sensors in vehicle location gives centimetric precision. 2003?

[9] M. Kais, S. Morin, A. De La Fortelle, and C. Laugier. Geometrical model to drive vision systems with error propagation. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, 2004.

[10] C. Pradalier and S. Sekhavat. Concurrent matching, localization and map building using invariant features. In *Proceedings of the*

IEEE International Conference on Intelligent Robots and Systems, 2002.

[11] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *IEEE International Conference on Robotics and Automation*, 2001.

[12] B. Steux, P. Coulombeau, and C. Laurgeau. Maps: a framework for prototyping automotive multi-sensor applications. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2000.

[13] R. Talluri and J. Aggarwal. Position estimation for an autonomous mobile robot in an outdoor environment. In *IEEE Transactions on Robotics and Automation*, 1992.

[14] C. Thorpe and J. Gowdy. Annotated maps for autonomous land vehicles. In *Proceedings DARPA Image Understanding Workshop*, 1990.

[15] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, University of North Carolina at Chapel Hill, 1995.