ROBOT POSITION ESTIMATION AND TRACKING USING SEQUENTIAL MONTE CARLO ALGORITHMS

Roberto Ferraz de Campos Filho, roberto.campos@poli.usp.br
Newton Maruyama, maruyama@usp.br
Jun Okamoto Junior, jokamoto@usp.br
Fabiano Rogerio Corrêa, fabiano_correa@yahoo.com
Escola Politécnica da Universidade de São Paulo

Fabio Kawaoka Takase, fktakase@mind.eng.br
Mind Open Source Technology

Abstract. *In this paper, we present a position measurement device for terrain mobile robots that describe planar movements. This measurement device makes use of a single uncalibrated camera taking pictures from a generic fixed position. The images projected on the image plane of the camera are rectified through a homography, which is a linear projective transformation. To perform this operation, the plane where the movement occurs must be known and at least four points from this plane on the camera image must be mapped to the corresponding four points on the rectified image. As the plane of the movement and the rectified image are parallel, measurements might be made. If at least one distance is known, the ratio between this real distance and the measured distance in the picture will be the same to all the points of the plane where the points are moving and real distances can be calculated. The visual tracking method is based on a Sequential Monte Carlo algorithm, commonly known as Particle Filter, that allows data fusion from different measurement sources. Two measurements are fused in this application: Color and Motion Cues. An experimental trial with a terrain mobile robot, the Pioneer P3-AT is presented. In order to evaluate the feasibility of the method, a comparison is made between position estimates provided by the measurement device and position estimates provided by the robot navigation system.*

Keywords: *Position Estimation, Tracking, Particle Filter, Mobile Robots*

1 INTRODUCTION

The aim of this work is the development of a position measurement sensor that is intended to be used with robots that make planar movements (*e.g.* terrain mobile robots, AUVs - Autonomous Underwater Vehicles - or ROVs - Remotely Operated Vehicles - in the surface of the water). To perform this position measurements, a visual tracking based on a Sequential Monte Carlo algorithm, also known as Particle Filter, is proposed. It is an external device that can be used for navigation by a robot, for example.

The camera is fixed in a generic position and real distances are obtained after the computation of a homography. This homography is the responsible for the perspective correction and it is computed by known points fixed in the movement plane. The 4 vertices of a rectangle is sufficient to compute this homography. The Hough Transformation (Duda and Hart, 1972) can also be used to detect this rectangle and find the vertices automatically, but it is not detailed in this paper.

Particle Filter is a popular tool to solve the tracking problems because it is simple and flexible. See examples of use in Doucet et al. (2001). An important advantage of the Particle Filter is that it allows data fusion from different sources. Two data sources are fused in this application: Color and Motion. The color localization cues are obtained by associating some reference color model with the object of interest and then compared with similar models extracted from each candidate position in the image. The tracking based only on color might present ambiguity in cases where other objects with the same reference color appear in the scene. The effect of this ambiguity is decreased fusing the color data with motion data, assuming that the tracked object is the only one that moves during the sequence of images. The motion localization cues are very similar to the color one. The difference is that the image used is the absolute difference between two consecutives images.

Various approaches have been developed to perform tracking algorithms. The CamShift (Bradski, 1998) is a fast color tracking algorithm that tracks position and size of an object. This is based on mean shift that is a popular method that finds the local maximum of a probability distribution (Cheng, 1995; Comaniciu and Meer, 1999). An approach proposed by Wang et al. (2009) uses the CamShift together with a particle filter and reduces significantly the amount of particles used to track. Blake and Isard (1997, 1998) perform visual tracking based on edge detection. Brasnett et al. (2007) develop a tracking system by fusing color, edge and texture cues using a particle filter approach.

The main contribution of the paper is the use of a tracking system in images rectified through a homography and also the performance comparison between three different movement models. As the images are rectified, the simple models developed has physical characteristics (cinematic models). Pérez et al. (2004) performs a model that represent a search in the image sequence. This model is implemented here as a Random Walk model and additionally two other models are also implemented (Constant Velocity and Constant Acceleration models).

The paper is organized as follows. Section 2 presents the standard Particle Filter and the data fusion scheme. Section 3 describes the algorithms that make the perspective correction and data fusion tracker based on color and motion. This section also presents motion models and a manner to weight the particles extracting information of color and motion from images, based on the work of Pérez et al. (2004, 2002). Section 4 presents and discuss a experiment with the terrain mobile robot Pioneer P3-AT and compare the trajectory computed by its sensors and the tracked through the image sequence.

2 PARTICLE FILTER

Particle Filter methods deal with the problem of estimating recursively the posterior distribution $p(x_t, y_{1:t})$, where x_t is the state of the system in current time step and $y_{1:t} = (y_1 \dots y_t)$ denotes all the observations up to the current time step. The basic idea of the method is to represent the probability density functions through samples (particles) and its respective weights (Blake and Isard, 1998; Gordon et al., 1993).

The system model is characterized as a *Hidden Markov Model* (HMM), the current state depends only on the previous state. Inside this context, the posterior distribution $p(x_t, y_{1:t})$ are computed in two steps, prediction and update:

1. Prediction step:

$$p(x_t|y_{1:t-1}) \quad = \quad \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{1}$$

2. Update step:

$$p(x_t|y_{1:t}) \quad \propto \quad p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{2}$$

The prediction step is obtained through marginalization and the update step through the Bayes' rule. The recursion requires a dynamic model $p(x_t|x_{t-1})$ and a model that gives the likelihood $p(y_t|x_t)$. In most cases, it is impossible to find an analytical solution to the prediction step, so representing the probability density functions as samples, the prediction step is obtained by passing the particles through the dynamic model $p(x_t|x_{t-1})$ (Schön, 2006).

According to Schön (2006), the Particle Filter algorithm is:

1. Initialize the particles, $\{x_{0|-1}^{(i)}\}_{i=1}^{M} \sim p_{x_0}(x_0)$ and set $t := 0$

2. Measurement update: calculate importance weights $\{q_t^{(i)}\}_{i=1}^{M}$ according to

$$q_t^{(i)} = p(y_t|x_{t|t-1}^{(i)}), i = 1, \dots, M \tag{3}$$

and normalize $\tilde{q}_t^{(i)} = \frac{q_t^{(i)}}{\sum_{j=1}^{M} q_t^{(j)}}$

3. Resampling: draw M particles, with replacement, according to

$$Pr(x_{t|t}^{(i)} = x_{t|t-1}^{(j)}) = \tilde{q}_t^{(j)}, i = 1, \dots, M \tag{4}$$

4. Time update: predict new particles according to

$$x_{t+1|t}^{(i)} \sim P(x_{t+1|t}|x_{t|t}^{(i)}), i = 1, \dots, M \tag{5}$$

5. Set $t := t + 1$ and iterate from step 2.

Using different kind of sensors give more robustness to the state computed by the algorithm. If one sensor fails or provides wrong data, others may correct the state estimation. Given M measurements, the measurement vector can be written as $y = (y^1 \dots y^M)$. Supposing the measurements are independent given the state, the likelihood can be factorized as:

$$p(y_t|x_t) = \prod_{m=1}^{M} p(y_t^m|x_t) \tag{6}$$

This implies that different kind of sensors needs its own model that gives the likelihood $p(y_t^m|x_t)$.

3 VISUAL TRACKING

This section describes each likelihood model (color and motion) used in the particle filter and the motion models necessary to the prediction step. First, the computation of the homography is introduced. The homography in necessary to take measurements directly from images or to correct the position extracted from the original ones.

3.1 Homography

In order to take measurements from a sequence of images, the pinhole camera model is used (Hartley and Zisserman, 2000). As the camera is fixed in a generic position, measurements cannot be taken directly from the images because the movement plane and the image plane are not parallel. On account of this, the original image is mapped to another image plane that is parallel to the movement plane. According to Criminisi et al. (1997), thus the corresponding points are related by:

$$\mathbf{X} = H\mathbf{x} \tag{7}$$

where H is a 3×3 homogeneous matrix, "=" is equality up to a scale factor and $\mathbf{x}$ and $\mathbf{X}$ are homogeneous 3-vector, $\mathbf{x} = (x, y, 1)$ and $\mathbf{X} = (X, Y, 1)$, that represent the original points in images and the mapped points, respectively.

In the presented application, the matrix H is constant because the camera is fixed. The plane to plane homography can be computed if points in the original image and the respective mapped points are known:

$$\begin{aligned} h_{11}x + h_{12}y + h_{13} &= h_{31}xX + h_{32}yX + h_{33}X \\ h_{21}x + h_{22}y + h_{23} &= h_{31}xY + h_{32}yY + h_{33}Y \end{aligned} \tag{8}$$

If 4 points and their correspondents are known (which 3 cannot be collinear), it is obtained 8 linear equations with 9 unknowns. As Eq. (7) is known up to a scale factor, one element of H can be fixed to compute the homography. Assuming $h_{33} = 1$, H can be computed. In this application, these 4 points are the vertices of a rectangle with known dimensions, so the mapped points form a rectangle proportional with the real one. This rectangle must appear in the first set of images and after computing the homography and rectifing the image, the tracking algorithm can be used. As the ratio between the real size of the rectangle and the size in pixels are known, each coordinate in the rectified image in pixels multiplied by this ratio gives the coordinates in real distances.

3.2 Color Cues

In order to track a target in a image sequence using the particle filter algorithm, it is necessary to fix a color likelihood model $p(y_t^C|x_t)$. The color based detection is made comparing a reference histogram (histogram of the known target) with the histogram of each particle. These histograms represents the color ranges in a region and what ranges are in more quantite of pixels. A histogram $h_x^c = (h_{1,x}^c \dots h_{B,x}^c)$, where $c \in \{RGB\}$, has B color range.

The distance between the histograms of each particle and the reference histogram is calculated using the Bhattacharyya similarity coefficient:

$$D(h_1, h_2) = \left(1 - \sum_{i=1}^{B} \sqrt{h_{i,1}h_{i,2}}\right)^{\frac{1}{2}} \tag{9}$$

where each histrogram needs to be normalized, i.e. $\sum_{i=1}^{B} h_{i,x}^c = 1$. Based on this distance, it is defined a color likelihood model as:

$$p(y^C|x) \quad \propto \quad exp\left(-\sum_{c\in\{RGB\}} \frac{D^2(h_x^c, h_{ref}^c)}{2\sigma_C^2}\right) \tag{10}$$

3.3 Motion Cues

The motion based detection is similar to the color based one. The diference is that the image used is the absolute diference between two consecutives images and the reference histogram is uniform:

$$h_{i,ref}^M \quad = \quad \frac{1}{B} \tag{11}$$

where $i = 1 \dots B$.

The histogram needs to be uniform to detect any variation on the image sequence. The region that the histograms are computed needs to be bigger than the others used in color likelihood model, because the image diference show the contour of the objects. Similarly of Eq. (10), it is defined a motion likelihood model as:

$$p(y^M|x) \quad \propto \quad exp\left(-\frac{D^2(h_x^M, h_{ref}^M)}{2\sigma_M^2}\right) \tag{12}$$

3.4 Motion Models

Three motion models were developed for this application: random walk (noise added in the position), constant velocity (noise added in the velocity components) and constant acceleration (noise added in the acceleration components).

The random walk model can be used to track even if the homography is unknown, because it is a simple search in a image sequence that depends only on the position in the image sequence (not the real one). These random walk model is constructed with a Gaussian Random Walk and a uniform component responsible to detect movements perceived as jumps in the image sequence. This motion model can be written as:

$$p(x_t|x_{t-1}) = (1 - \beta_u)N(x_t|x_{t-1}, \Lambda) + \beta_u U_\chi(x_t) \tag{13}$$

where $N(.|\mu, \Lambda)$ is a Gaussian distribution with mean μ and covariance Λ, $U_A(.)$ is a uniform distribution over a set A, $0 \le \beta_u \le 1$ is the weight of the uniform component.

The other two models depend on the real velocities and accelerations of the object tracked, so the homography needs to be known. The constant velocity model is:

$$\begin{aligned} x_{t+1} &= x_t + \dot{x}_t \cdot t \\ \dot{x}_{t+1} &= \dot{x}_t + e_t \end{aligned} \tag{14}$$

and the constant acceleration model is:

$$\begin{aligned} x_{t+1} &= x_t + \dot{x}_t \cdot t + \frac{\ddot{x}_t \cdot t^2}{2} \\ \dot{x}_{t+1} &= \dot{x}_t + \ddot{x}_t \cdot t \\ \ddot{x}_{t+1} &= \ddot{x}_t + e_t \end{aligned} \tag{15}$$

where e_t is a Gaussian noise. These equations are applied in y direction similarly.

The aim of these models is to detect fast movements, i.e. when the random walk model fails.

4 EXPERIMENTAL TRIALS

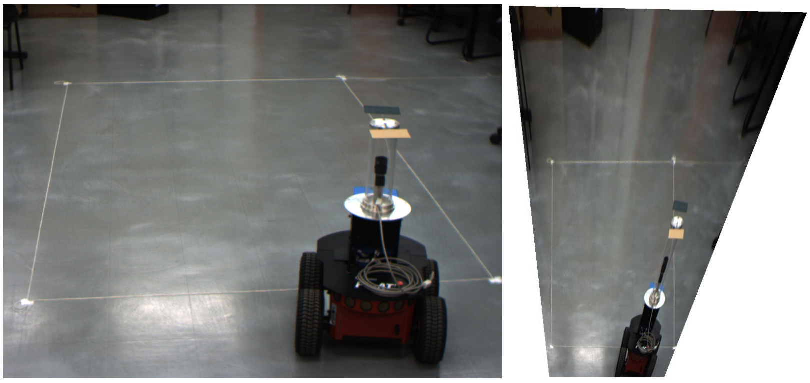
The robot Pioneer P3-AT was used to make some comparisons between the motion models of the position sensor developed. There are some embedded sensors in this robot that permit the computation of the trajectory made by it, but it is not the scope of this paper discuss how this estimation is made by the robot. The trajectory computed by the robot is precise enough to be assumed as the real one. This position sensor was developed to detect position (x, y) in images, so it is used two landmarks on the top of the robot to detect its position and rotation (x, y, θ). The tracking of these landmarks occurs in independent way and the rotation is calculated through these two points.

Figure 1 shows the perspective correction procedure detailed in section 3.1. The rectangle is used to compute the homography and the tracking algorithm is used after applying the homography in each image. As can be seen in Fig. 1, the plane where the homography was computed is not the same plane where the tracking occurs. So the coordinates must be corrected geometrically:

$$x' = x\frac{h}{h - z} \tag{16}$$

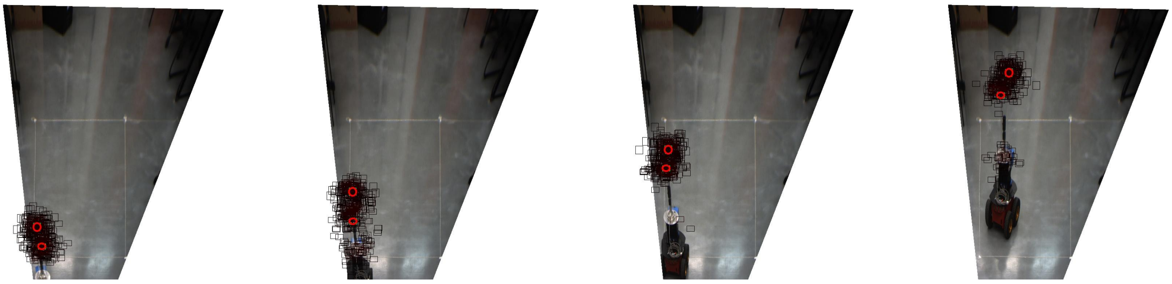
where x is the coordinate in the rectified image, h is the camera height and z is the robot height.

Figure 2 shows some scenes of the tracking and Fig. 3 shows the difference between the three models and the real trajectory. In order to make a quantitative error analysis, the distance between the "real" position and the estimated one by the position sensor in the same time t is used to compute the mean error and its deviation (see Tab. 1).

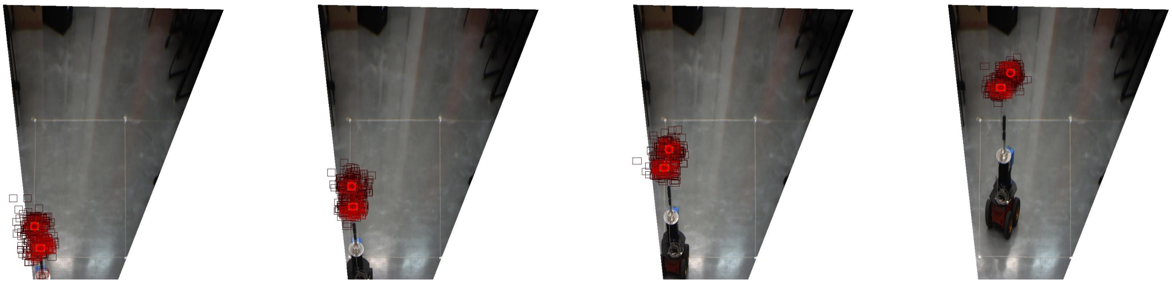


(a) (b)

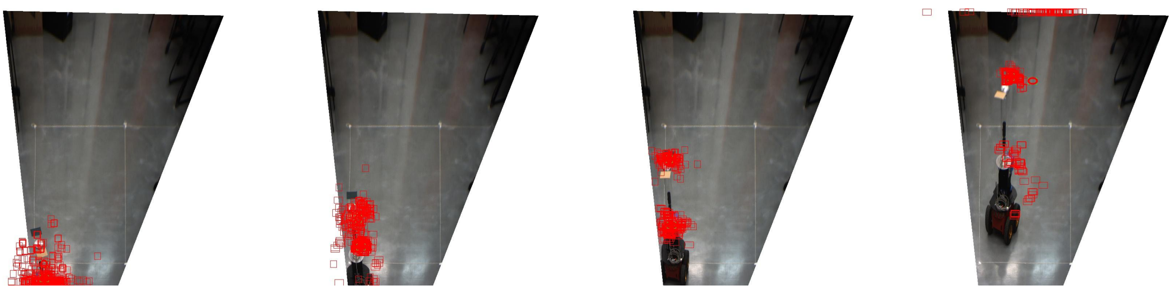
Figure 1. The figure 1(a) is the original one and figure 1(b) is the retificated one. The rectangle with known dimensions is used to compute the homography



(a) Random Walk model



(b) Constant Velocity model



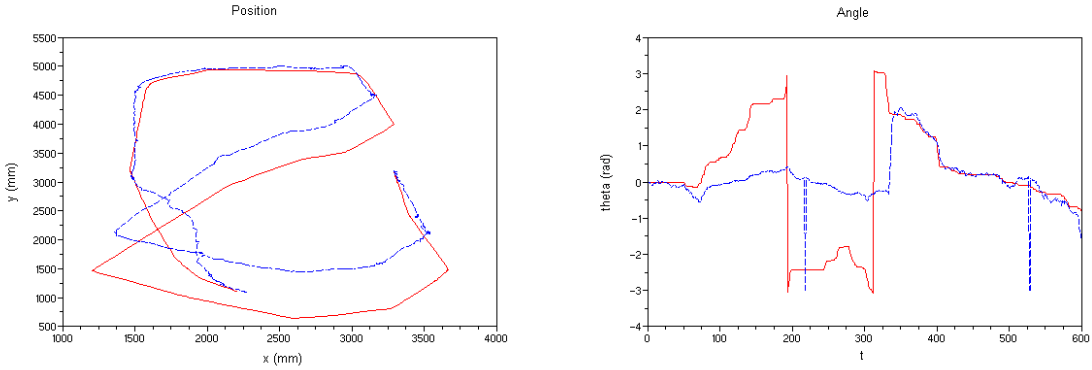
(c) Constant Acceleration model

Figure 2. Tracking using the fusion between color and motion with Random Walk model, Constant Velocity model and Constant Acceleration model

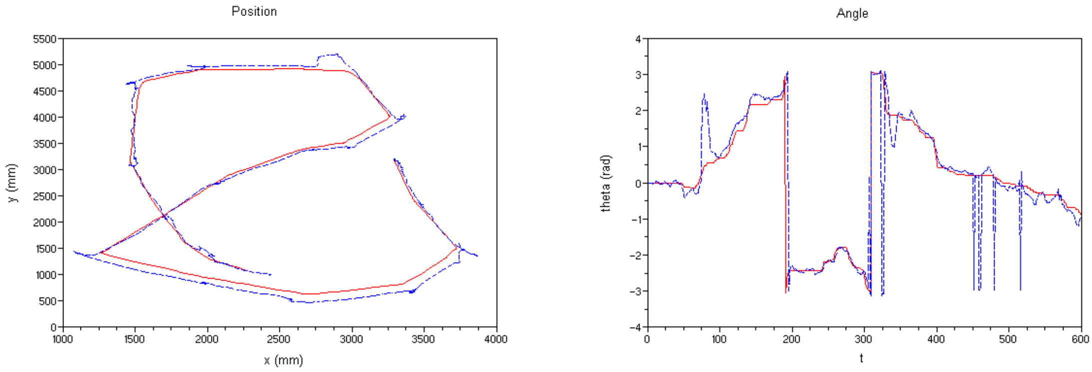
Table 1. Error Analysis of the Position Sensor

Model	Mean Error$^{(1)}$ (cm)	Deviation (cm)
Random Walk	28.0	23.3
Constant Velocity	13.0	8.1
Constant Acceleration	106.2	82.8

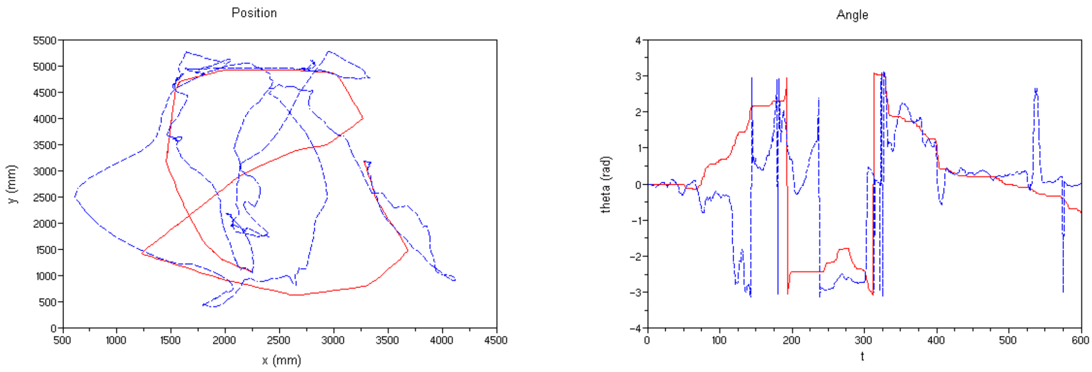
$^{(1)}$: $error(t) = dist(x(t)_{real} - x(t)_{positionsensor})$



(a) Random Walk model



(b) Constant Velocity model



(c) Constant Acceleration model

Figure 3. Comparison between the three models (blue line) and the real trajectory (red line)

5 CONCLUSION

In this paper we have presented the basic mechanisms of the Particle Filter algorithm and a tracking application has been developed fusing datas from color and motion cues. The objective has been tracking objects and getting coordinates in real world. In order to obtain the coordinates measurements, a perspective correction procedure has been introduced. Three motion models have been developed and tested with the terrain mobile robot Pioneer P3-AT.

The constant velocity model has been the one that presented better results, as can be seen in Fig. 2 and Fig. 3 and verified through Tab. 1. The random walk model can also be used in this application, but it fails when the tracked object moves faster and it has been observed a slower response than the constant velocity model. The constant acceleration model loses the tracked object easily because the noise propagation to the position increases a lot and the state variable weighted are the position (x, y). A fusion with velocity or acceleration sensors could improve these model, because the particles would have more variables of the state vector weighted.

6 ACKNOWLEDGEMENTS

Authors would like to acknowledge the Fapesp (Proc. n^o 2007/00404-3) for the scholarship support of the first author and the CNPq (Proc. n^o 550934/2005-7) for the financial support.

7 REFERENCES

Blake, A. and Isard, M., 1997, “Active Contours”, Springer-Verlag.

Blake, A. and Isard, M., 1998, “CONDENSATION - conditional density propagation for visual tracking”, Vol. 29, pp. 5–28.

Bradski, G. R., 1998, “Computer Vision Face Tracking For Use in a Perceptual User Interface”.

Brasnett, P., Mihaylova, L., Bull, D. and Canagarajah, N., 2007, “Sequential Monte Carlo Tracking by Fusing Multiple Cues in Video Sequences”, Vol. 25, pp. 1217–1227.

Cheng, Y., 1995, “Mean shift, mode seeking, and clustering”, Vol. 17, pp. 790–799.

Comaniciu, D. and Meer, P., 1999, “Mean shift analysis and applications”.

Criminisi, A., Reid, I. and Zisserman, A., 1997, “A Plane Measuring Device”.

Doucet, A., Freitas, N. D. and Gordon, N., editors, 2001, “Sequential Monte Carlo Methods in Practice (Statistics for Engineering and Information Science)”, Springer-Verlag, New York.

Duda, R. O. and Hart, P. E., 1972, “Use of the Hough Transformation To Detect Lines and Curves in Pictures”.

Gordon, N. J., Salmond, D. J. and Smith, A. F. M., 1993, “Novel approach to nonlinear and non-Gaussian Bayesian state estimation”, Vol. 140, pp. 107–113.

Hartley, R. and Zisserman, A., 2000, “Multiple View Geometry in Computer Vision”, Cambridge University Press.

Pérez, P., Hue, C., Vermaak, J. and Gangnet, M., 2002, “Color-Based Probabilistic Tracking”, pp. 661–675.

Pérez, P., Vermaak, J. and Blake, A., 2004, “Data Fusion for Visual Tracking with Particles”, pp. 495–513.

Schön, T. B., 2006, “Estimation of Nonlinear Dynamic Systems - Theory and Applications”, PhD thesis, Linköpings universitet, Linköping, Sweden.

Wang, Z., Yang, X., Xu, Y. and Yu, S., 2009, “CamShift guided particle filter for visual tracking”, Vol. 30, pp. 407–413.

8 Responsibility notice

The authors are the only responsible for the printed material included in this paper.