CONSISTENCY CHECKS FOR PARTICLE FILTERS WITH APPLICATION TO IMAGE STABILIZATION

Ferdinand van der Heijden

F.vanderHeijden@utwente.nl
University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science, Enschede, The Netherlands

ABSTRACT

An 'inconsistent' particle filter produces – in a statistical sense – larger estimation errors than predicted by the model on which the filter is based. Inconsistent behavior of a particle filter can be detected online by checking whether the predicted measurements (derived from the particles that represent the one-step-ahead prediction pdf) comply in a statistical sense with the observed measurements.

This principle is demonstrated in an image stabilization application. We consider an image sequence of a scene consisting of a dynamic foreground and a static background. The motion of the camera (slow rotations and zooming) is modeled with an 8-dim state vector describing a projective geometrical transformation that, inversely applied to the current frame, compensates the camera motion. The dynamics of the state vector is modeled as a first order AR process.

The measurements of the system are corner points (detected in the first frame) that are tracked. The particle filtering estimates the state vector using the measurements. However, the filter behaves inconsistently because a few corner points belong to the foreground. Using inconsistency checks these foreground points are detected and removed from the list of measurements.

1. INTRODUCTION

In the last decade, a new methodology for state estimation has emerged. 'Particle filtering' uses a Monte Carlo approach to represent the probability densities of the underlying process [1], [2]. A set of samples is able to represent any probability density. As such, particle filtering has the potential to cope with non-Gaussian and nonlinear cases, as well as with discrete cases, i.e. hidden Markov models, and even with mixed cases (e.g. continuous states interacting in different modes).

An important aspect of the design of a particle filter is the selection of a suitable model that describes the underlying physical process and the sensory system. Errors in the model inevitable lead to estimation errors. The model should be accurate enough so that the influence of modeling errors is just negligible compared with the errors due to process noise and measurement noise. On the other hand, the model should not be too fine because in that case many parameters must be determined during the stage of system identification. An overkill of parameters easily leads to overfitting. Such a system may also be sensitive to small deviations of the parameters.

The term 'inconsistency' refers to a situation in which a particle filter produces estimation errors that – in a statistical sense – are larger than predicted by the model on which the filter is based. Inconsistent behavior is caused by two possible effects. First, the number of samples may be too small to represent the probability densities with sufficient statistical significance. The second cause is the modeling errors mentioned above. If these errors are too large, then the estimates will be biased.

In this paper a set of statistical test variables will be defined that are useful to detect whether in a given particle filter the model being used is accurate enough with respect to the uncertainties due to process noise and measurement noise. The test variables are also useful for the detection of inconsistencies of a particle filter during its online operation. The purpose then is to test whether the particle filter is still operating properly. As such these test variables are useful for fault detection, and for the detection of drifting parameters, and so on.

The usefulness of the proposed test variables is demonstrated in an image processing application: the stabilization of an image sequence. We consider an image sequence of a scene consisting of a large static background and a dynamic foreground. In the image sequence the background is moving caused by rotations and zooming of the camera. The purpose is to undo the image motion caused by the camera movements.

Section 2 is confined to the theoretical part of the paper. After an introduction summarizing the particle filter the test variables are defined. Section 3 is a demonstration of these test variables applied in the image stabilization application. The paper finalizes with a short conclusion.

2. CONSISTENCY CHECKS IN PARTICLE FILTERING

This section first summarizes the particle filtering and the underlying model (also introducing the notation), and then defines a set of test variables that are useful to detect inconsistent behavior of the filter.

2.1. Particle filtering

The state model that is considered in this paper consists of a state space model of the physical process:

$$\mathbf{x}(i+1) = \mathbf{f}\left(\mathbf{x}(i), \mathbf{w}(i)\right) \tag{1}$$

$$\mathbf{z}(i) = \mathbf{h}\left(\mathbf{x}(i), \mathbf{v}(i)\right) \tag{2}$$

i is the discrete time index. $\mathbf{x}(i)$ is the state vector. $\mathbf{f}()$ is the state equation. The process noise, $\mathbf{w}(i)$, is an independent noise sequence statistically defined by the pdf $p_{\mathbf{w}}(\mathbf{w})$. $\mathbf{z}$ is the M-dimensional measurement vector. $\mathbf{h}()$ is the measuremetn function which models the sensory system. The sensor noise, $\mathbf{v}(i)$, is an independent noise sequence with pdf $p_{\mathbf{v}}(\mathbf{v})$.

The state equation, together with $p_{\mathbf{w}}(\mathbf{w})$, defines a transition pdf $p_t\left(\mathbf{x}(i+1) \mid \mathbf{x}(i)\right)$. Taken together, $\mathbf{h}()$ and $p_{\mathbf{v}}(\mathbf{v})$ define the conditional pdf $p_{\mathbf{z}}\left(\mathbf{z}(i) \mid \mathbf{x}(i)\right)$. Starting from the prior pdf $p_0\left(\mathbf{x}(0)\right)$ of $\mathbf{x}(0)$ the online posterior pdf is iteratively obtained using the following relations:

$$\begin{aligned} p\left(\mathbf{x}(i) \mid \mathbf{Z}(i)\right) &= \frac{1}{c}\, p_{\mathbf{z}}\left(\mathbf{z}(i) \mid \mathbf{x}(i)\right) p\left(\mathbf{x}(i) \mid \mathbf{Z}(i-1)\right) \\ p\left(\mathbf{x}(i+1) \mid \mathbf{Z}(i)\right) &= \int_{\mathbf{x}(i)\in X} p_t\left(\mathbf{x}(i+1) \mid \mathbf{x}(i)\right) p\left(\mathbf{x}(i) \mid \mathbf{Z}(i)\right) d\mathbf{x}(i) \end{aligned} \tag{3}$$

c is a normalization constant. $\mathbf{Z}(i)$ is the sequence $\{\mathbf{z}(0), \cdots, \mathbf{z}(i)\}$ of available measurements.

The particle filter keeps a representation of $p\left(\mathbf{x}(i) \mid \mathbf{Z}(i)\right)$ by means of a set of weighed samples, the particles $\{\mathbf{x}_s^{(n)}(i), w^{(n)}(i)\}$ with $n = 1, \cdots, N$ and $\sum_{n=1}^{N} w^{(n)}(i) = 1$. An optimal estimate $\hat{\mathbf{x}}(i)$ is obtained by applying some optimality criterion, e.g. minimum mean square error, maximum a posteriori, etc., and finding the $\mathbf{x}(i)$ that maximizes that criterion. Expectations are approximated by

$$\mathrm{E}[g(\mathbf{x}(i)) \mid \mathbf{Z}(i)] \approx \sum_{n=1}^{N} w^{(n)}(i) g(\mathbf{x}_s^{(n)}(i)) \tag{4}$$

The *n*-th sample $\mathbf{x}_s^{(n)}(i)$ is drawn from a so-called proposal density function $q(\mathbf{x}(i) \mid \mathbf{x}_s^{(n)}(i-1), \mathbf{z}(i))$. A popular choice is to define:

$$q(\mathbf{x}(i) \mid \mathbf{x}_s^{(n)}(i-1), \mathbf{z}(i)) \stackrel{def}{=} p_t(\mathbf{x}(i) \mid \mathbf{x}_s^{(n)}(i-1)) \tag{5}$$

Now and then, a particle filter needs resampling, i.e. the act of redrawing samples from another density with the goal to equalize the weights. A popular method is systematic resampling which redraws samples from the density

$$p(\mathbf{x}) = \sum_{n=1}^{N} w^{(n)}(i) \delta\left(\mathbf{x} - \mathbf{x}_s^{(n)}(i)\right) \tag{6}$$

The new samples are copies of some of the old samples, but with multiplicities that are proportional to their old weights. The new weights are reset to $1/N$. A single iteration step of the used particle filter is as follows.

$$\begin{aligned} &1.\quad \text{for } n = 1:N \\ &\qquad 1.1\quad \text{draw } \mathbf{x}_s^{(n)}(i) \text{ from } p_t\left(\mathbf{x}(i) \mid \mathbf{x}_s^{(n)}(i-1)\right) \\ &\qquad 1.2\quad w^{(n)}(i) = p_{\mathbf{z}}\left(\mathbf{z}(i) \mid \mathbf{x}_s^{(n)}(i)\right) \\ &2.\quad \text{normalize:}\ \ w^{(n)}(1:N) := w^{(n)}(1:N) \,/ \sum w^{(n)}(1:N) \\ &3.\quad \text{resample} \end{aligned} \tag{7}$$

This type of particle filtering is called SIR filtering (Sampling Importance Resampling) [2].

2.2. Test variables for consistency checks

The purpose of this section is to find a concept which allows us to check online whether the behavior is (still) consistent, i.e. whether the effects of modeling errors are not too severe. The principle is that using (1) and (2) we determine the pdf $p(\mathbf{z}(i) \mid \mathbf{Z}(i-1))$ and check whether the observed $\mathbf{z}(i)$ complies with this pdf. Suppose that, using all previous measurements $\mathbf{Z}(i-1)$ up to time $i-1$, the pdf of the state $\mathbf{x}(i)$ is $p\left(\mathbf{x}(i) \mid \mathbf{Z}(i-1)\right)$. Then the probability of $\mathbf{z}(i)$ is:

$$\begin{aligned} p\left(\mathbf{z}(i) \mid \mathbf{Z}(i-1)\right) &= \int_{\mathbf{x}} p\left(\mathbf{z}(i), \mathbf{x}(i) \mid \mathbf{Z}(i-1)\right) d\mathbf{x} \\ &= \int_{\mathbf{x}} p_{\mathbf{z}}\left(\mathbf{z}(i) \mid \mathbf{x}(i)\right) p\left(\mathbf{x}(i) \mid \mathbf{Z}(i-1)\right) d\mathbf{x} \end{aligned} \tag{8}$$

The pdf $p\left(\mathbf{x}(i) \mid \mathbf{Z}(i-1)\right)$ is represented by the predicted samples. Therefore, using (8), the pdf $p\left(\mathbf{z}(i) \mid \mathbf{Z}(i-1)\right)$ can be calculated online. The filter is consistent only if the sequence of observed measurements $\mathbf{z}(i)$ obeys the statistical properties prescribed by the sequence of pdfs $p\left(\mathbf{z}(i) \mid \mathbf{Z}(i-1)\right)$.

A test whether all $\mathbf{z}(i)$ comply with $p\left(\mathbf{z}(i) \mid \mathbf{Z}(i-1)\right)$ is not easy, because this pdf depends on i. The problem will be tackled by treating each scalar measurement separately. Consider the m-th element $z_m(i)$ of the measurement vector, and assume that $p_m(z, i)$ is its hypothesized marginal probability density. We assume that $z_m(i)$ has a continuous distribution. Suppose that the cumulative distribution of $z_m(i)$ is

$$F_m(z, i) = \int_{-\infty}^{z} p_m(\zeta, i) d\zeta \tag{9}$$

Then the random variable $u_m(i) = F_m(z_m(i), i)$ has a uniform distribution between 0 and 1 [3]. The consistency check boils down to testing whether the set of $u_m(i)$, calculated over the past *I* moments of time, i.e. $\{u_m(j) \mid j \in MW(i) \text{ and } m = 1, \cdots, M\}$, indeed has such a uniform distribution. $MW(i)$ is a moving window that contains the recent *I* past time indices, i.e. $MW(i) = \{i - I + 1, \cdots, i\}$.

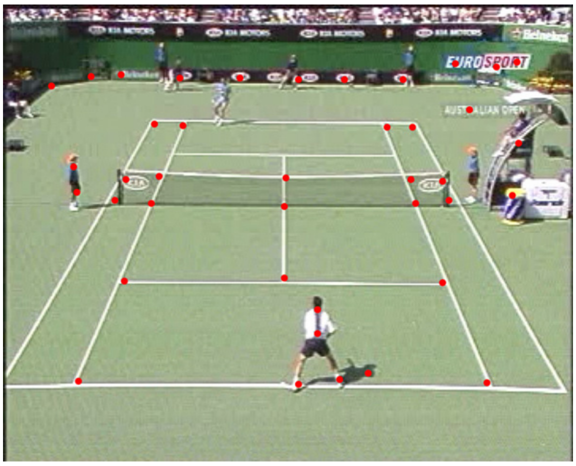

Figure 1. *The first frame of a video clip of a tennis play together with Harris corner points.*

In the SIR filter, the pdf $p\left(\mathbf{x}(i) \mid \mathbf{Z}(i-1)\right)$ is represented by the unweighted samples $\mathbf{x}_s^{(n)}(i)$. The pdf $p\left(\mathbf{z}(i) \mid \mathbf{Z}(i-1)\right)$ can then be represented by *N* samples $\mathbf{z}_s^{(n)}(i)$ drawn from $p_{\mathbf{z}}\left(\mathbf{z}(i) \mid \mathbf{x}_s^{(n)}(i)\right)$. The variables $u_m(i)$ are obtained next by counting the number of samples $\mathbf{z}_s^{(n)}(i)$ for which the *m*-th element is smaller or equal to the *m*-th element of the observed $\mathbf{z}(i)$. This count should be normalized by *N* to get the relative count. Thus, if $z_{s,m}^{(n)}(i)$ is the *m*-th element from $\mathbf{z}_s^{(n)}(i)$, and $z_m(i)$ the *m*-th element from $\mathbf{z}(i)$, then:

$$u_m(i) = \frac{1}{N}\sum_{n=1}^{N}\left(z_{s,m}^{(n)}(i) \le z_m(i)\right) \qquad \text{for} \quad m = 1,\cdots,M \tag{10}$$

This is a cheap computational step that should be performed immediately after step 1.1 in the algorithm shown in (7).

3. IMAGE STABILIZATION

The objective of image stabilization is to undo the effects of the ego-motion in an image sequence, i.e. the effects due to camera motion. We consider applications in which the scene consists of a static background and a dynamic foreground. The purpose of the image stabilization is to fix the background image. A typical application is the analysis of a tennis game (Figure 1) based on TV broadcasted video clips. Such an analysis is facilitated by background detection which is easy if the background is stationary.
The problem of image stabilization can be solved in various ways. A popular technique is to localize features in one or more image frames and to track them during their lifetime in consecutive frames See [4] and the references therein. A simple method is to use low level point features (such as Harris corners, or Susan corners). The advantage is that these features do not depend much on the application (in contrast with high level features), but here the weakness is that some of these features may belong to the foreground, and thus outlier detection is needed.

3.1. The model

We use an 8-D state space model to describe the effect of the camera motion. The state vector defines a projective geometrical transform that explains how the background of the i-th frame of sequence can be derived from the background of the first frame. The projective transform is uniquely defined by the four corner points of the image frame, expressed in the image coordinate system of the first frame. In that case, a number of 8 decoupled first order AR processes are suitable to describe the sequence of geometrical transforms:

$$\mathbf{x}(i+1) = \mathbf{x}(i) + \mathbf{w}(i) \text{ with } \mathbf{C}_w = \sigma_w^2\mathbf{I} \tag{11}$$

Eq. (11) is a white-noise displacement model. It simply states that the difference of the corner positions between two consecutive frames has a zero-mean Gaussian distribution with a standard deviation of σ_w pixels. Since σ_w is kept small (e.g. $\sigma_w = 2$) the model only allows a smooth sequence of geometrical transforms. The initial condition of the state equation is that $\mathbf{x}(0)$ equals the corner position of the first image.
The measurement system that we use consists of an Harris corner detector [5] applied to the first frame, followed by a Lucas-Kanade tracker [6] that estimates the detected corners in the consecutive frames. Suppose that $\mathbf{z}(i)$ is the sequence of coordinates of corners, then the nonlinear measurement model is:

$$\mathbf{z}(i) = T(\mathbf{x}(i))\mathbf{z}(0) + \mathbf{v}(i) \tag{12}$$

where $T(\mathbf{x}(i))$ is the projective transform that relates the i-th frame to the first frame. $\mathbf{v}(i)$ is the sensor noise which models the estimation error of the Lucas-Kanade tracker. We simply assume zero-mean, white Gaussian noise with covariance matrix $\mathbf{C}_\mathbf{v} = \sigma_v^2\mathbf{I}$. The standard deviation of the noise is set to $\sigma_\mathbf{v} = 2$ pixels.

3.2. Selection of corners using consistency checks

The corners shown in Figure 1 belong to two possible classes: background corners and foreground corners. Only the background corners are of use. The particle filter estimates the corners $\mathbf{x}(i)$ of the image frame. Next, this estimate is used to apply an inverse transform. In the ideal case, all Harris background corners will be mapped back to their original position in the first frame. Unfortunately, the motion of the foreground corners is

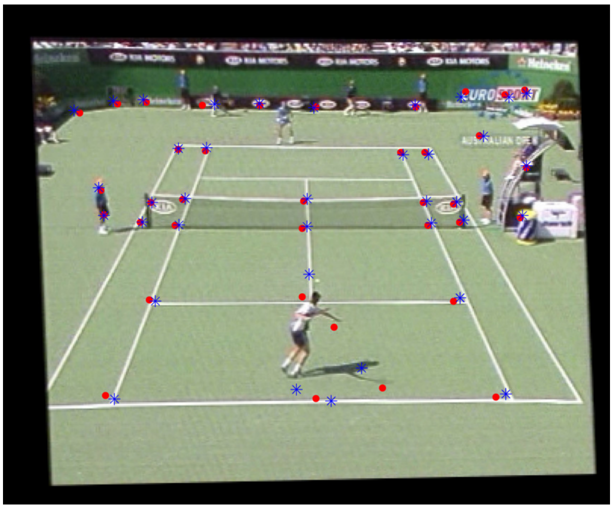

Figure 2. *Frame 22 after stabilization using all 36 corners.*

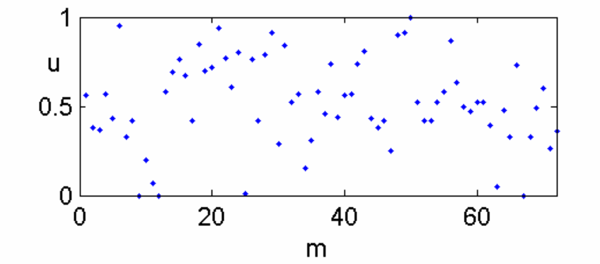

Figure 3. *The 72 test variables for frame 22.*

not consistent with the background motion, and therefore, the back-mapped Harris corners do not coincide with the original one as Figure 2 shows (here, the back-mapped, tracked features are represented by the blue '*').

Consistency checks must be used to filter out the foreground corners. The exact way to use the test variables $u_m(i)$ is to store the last I values in a FIFO buffer and apply a distribution test to see whether they comply with a uniform distribution. Unfortunately, the number of frames is too small for such a statistical analysis. Instead we use the heuristic rule that all Harris corners for which $u_m(i)$ is driven to the extreme, e.g. $u_m(i) > 0.99$ or $u_m(i) < 0.01$ during the last few time step is marked as inconsistent, and removed from the list. Figure 3 shows the 72 variables corresponding to the 36 corners in Figure 2. Here, four corners appear to be candidate foreground corners. Figure 4 shows the same frame, but now with the four corners removed from the list. Clearly, the remaining back-mapped Harris corner points now fits the original Harris points.

4. CONCLUSION

We have introduced test variables that can be used to

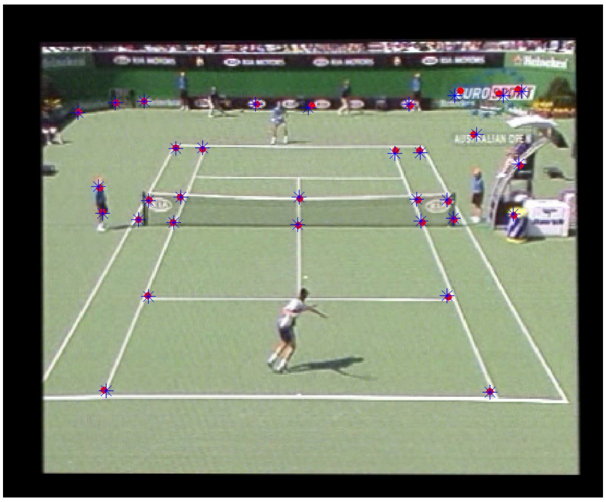

Figure 4. *Frame 22 after stabilization using only the selected corners.*

detect inconsistent behavior of a particle filter. The test variables are derived from a particle representation of the pdf of the measurement vector given all previous measurement vectors. The observed measurement vector must comply with this pdf.

We have demonstrated the usefulness of the test variables in an image stabilization application where some inconsistent measurements coming from foreground pixels interferes with measurements coming from the background.

5. REFERENCES

[1] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/nonGaussian Bayesian state estimation, IEE Proceedings-F 140(2), pp 107-113, 1993.

[2] M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, IEEE Transactions on Signal Processing, Vol. 50, No. 2, pp. 174-188, February 2002.

[3] M. Rosenblatt, Remarks on a multivariate transformation, Annals of Mathematical Statistics, Vol. 23, No. 3, pp 470-472, September 1952.

[4] B. Jung and G.S. Sukhatme, Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment,The 8th Conference on Intelligent Autonomous Systems, pp. 980--987, Amsterdam, The Netherlands, March 10-13, 2004.

[5] C. J. Harris and M. Stephens. A combined corner and edge detector. In Proc. 4th Alvey Vision Conf., Manchester, pages 147-151, 1988.

[6] B. Lucas, T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of Imaging Understanding Workshop, pp. 121-130 (1981).