Stochastic complexity integral image based technique for fast video tracking

Jean-François Boulanger,[1,2] Frédéric Galland,[1,*] Pascal Martin,[2] and Philippe Réfrégier[1]

[1]*Institut Fresnel, Aix-Marseille Université, Ecole Centrale Marseille, CNRS, Campus de Saint Jérôme, 13013 Marseille, France*

[2]*Kaolab, Tech Indus D, 645 Avenue Mayor de Montricher, Pôle d'activités d'Aix les Milles, 13854 Aix-en-Provence Cedex 3, France*

**Corresponding author: frederic.galland@fresnel.fr*

Received July 22, 2008; accepted September 17, 2008;
posted October 3, 2008 (Doc. ID 99189); published October 28, 2008

We propose a new method based on the minimization of the stochastic complexity for fast and efficient tracking adapted to video images with a static camera. The obtained criterion combines the advantages of background-subtraction-based techniques and those of using measures of similarities to a target model without requiring any tuning of a weighting parameter. It is then demonstrated that this approach can be implemented with a fast integral image technique to estimate the location and the rectangular shape of the target in a few milliseconds. © 2008 Optical Society of America

OCIS codes: 100.4999, 100.2000, 110.4280.

Tracking in video image sequences is a key point in many applications and particularly in video surveillance. Owing to real-time constraints, fast and efficient solutions are required. Moreover, in the general case, the tracked target can evolve in depth and is nonrigid. Therefore, not only the target location but also its size have to be estimated. In this Letter, a new technique based on the minimization of the stochastic complexity [1] and on an integral image [2] implementation is proposed. It is shown that it allows one to perform target tracking very quickly without requiring assumptions on the target size or embedded parameters that need to be tuned by the user, and with simple probability density functions (PDFs) for the background and target gray levels. The particular case of a static camera is studied, which makes it possible to model the fluctuations of the background [3] during a learning period. This assumption is combined with the knowledge of the target gray levels extracted from the previous frames.

Let $b^t(x,y)$ be the background value in pixel (x,y) and at time t. In the following, $b^t(x,y)$ is assumed to be the realization of a random stationary variable with a PDF $P^B_{(x,y)}$ that depends on the coordinates of the considered pixel. This PDF $P^B_{(x,y)}$ can be estimated during a calibration step or online. Let s^t be the image at time t. Background subtraction approaches can be implemented [3–5] to detect whether or not the pixel value $s^t(x,y)$ should be considered as a realization of the background PDF $P^B_{(x,y)}$. Blob-matching techniques [3] are then required to recover the target location from this binary detection map. Although these techniques can lead to efficient tracking algorithms, the main limitation is that tracking and detection processes are performed independently.

Other approaches were proposed [6–8] that directly try to recover a known target in the image. Generally these approaches consist of finding the target location that optimizes a measure of similarity between the features extracted from target candidates at time t and the expected features of the target. Nevertheless, these approaches do not take benefit from the learned background PDF $P^B_{(x,y)}$. Moreover, as demonstrated in [9], the estimation of the target size cannot be directly addressed with such approaches without strong regularization.

The method proposed in this Letter takes into account that the camera is fixed and that the radiometry of the target is partially known. As shown in the following, these two hypotheses allow one to cope with the aforementioned limitations.

Let Ω denote a set of pixels that correspond to a candidate target, and let Ω^C be the complementary region of Ω in s^t. The stochastic complexity $\Delta^t(\Omega)$ of the image s^t is defined as the code length needed to encode s^t with entropic codes and with a given candidate target region Ω. This code length is the sum of three terms:

$$\Delta^t(\Omega) = \Delta^t_{\text{shape}} + \Delta^t_{\Omega} + \Delta^t_{\Omega^C}. \tag{1}$$

The first term Δ^t_{shape} corresponds to the encoding of the target shape Ω, and the two others, denoted Δ^t_{Ω} and $\Delta^t_{\Omega^C}$, correspond to the encoding of the gray level values of the pixels inside Ω and Ω^C, respectively. Since the camera is assumed to be fixed, it is useless to directly encode the pixel values inside Ω^C, but only their fluctuations with respect to the previously learned background model $P^B_{(x,y)}$ that are assumed spatially independent. In this case, according to [1,10], the code length necessary to encode the pixel value $s^t(x,y)$ [with $(x,y) \in \Omega^C$] can be approximated by its Shannon quantity of information, i.e., $\log P^B_{(x,y)}(s^t(x,y))$, leading to

$$\Delta^t_{\Omega^C} = -\sum_{(x,y)\in\Omega^C} \log P^B_{(x,y)}(s^t(x,y)). \tag{2}$$

On the contrary, inside Ω, the pixel values are supposed to be in adequacy with the target model. Let P^A

denote the target PDF (identical for the whole pixels of the target) that has been estimated in previous images. Similarly, the code length required to encode the pixel values inside Ω is thus

$$\Delta_{\Omega}^{t} = -\sum_{(x,y)\in\Omega} \log P^{A}(s^{t}(x,y)). \tag{3}$$

For the sake of computational cost reduction, the PDFs P^{A} and $P^{B}_{(x,y)}$ are assumed to be Gaussian PDFs with mean and variance equal to m_A and σ_A^2 for the target and equal to $m_B(x,y)$ and $\sigma_B^2(x,y)$ for the background. It will be shown in the following that the proposed stochastic complexity framework allows one to obtain good results with such a simple modelization.

Equations (2) and (3) can then be directly rewritten as

$$\Delta_{\Omega}^{t} = \frac{N_{\Omega}}{2}\log 2\pi + \frac{N_{\Omega}}{2}\log\sigma_A^2 + \frac{1}{2}\sum_{(x,y)\in\Omega} f_A^t(x,y),$$

$$\Delta_{\Omega^C}^{t} = \frac{N_{\Omega^C}}{2}\log 2\pi + \frac{1}{2}\sum_{(x,y)\in\Omega^C} f_B^t(x,y), \tag{4}$$

with N_{Ω} and $N_{\Omega C}$ as the pixel number inside Ω and Ω^C and with

$$f_A^t(x,y) = \frac{(s^t(x,y)-m_A)^2}{\sigma_A^2},$$

$$f_B^t(x,y) = \log\sigma_B^2(x,y) + \frac{(s^t(x,y)-m_B(x,y))^2}{\sigma_B^2(x,y)}. \tag{5}$$

To estimate very quickly the size and location of the target, it is proposed to model the target shape with a rectangle (with horizontal and vertical directions). In this case, the shape Ω can be encoded by providing the coordinates of two opposite nodes of the rectangle, which leads to a code length independent on Ω. The minimization of the stochastic complexity $\Delta^t(\Omega)$ is then equivalent to the minimization of $\Delta_{\Omega}^{t}+\Delta_{\Omega^C}^{t}$, i.e., the estimated target shape $\tilde{\Omega}^t$ at time t is the shape that minimizes $\Delta_{\Omega}^{t}+\Delta_{\Omega^C}^{t}$.

When defining

$$K^t = \frac{1}{2}\sum_{(x,y)\in\text{Image}} f_B^t(x,y), \tag{6}$$

the expression of $\Delta_{\Omega^C}^{t}$ [Eq. (4)] can be rewritten as

$$\Delta_{\Omega^C}^{t} = K^t + \frac{N_{\Omega^C}}{2}\log 2\pi - \frac{1}{2}\sum_{(x,y)\in\Omega} f_B^t(x,y), \tag{7}$$

leading to

$$\Delta_{\Omega}^{t}+\Delta_{\Omega^C}^{t} = K^t + \frac{N}{2}\log 2\pi + \frac{N_{\Omega}}{2}\log\sigma_A^2 + \frac{1}{2}\sum_{(x,y)\in\Omega}[f_A^t(x,y)-f_B^t(x,y)], \tag{8}$$

where $N=N_{\Omega}+N_{\Omega C}$ is the number of pixels in the whole image. Since K^t and $\frac{N}{2}\log 2\pi$ do not depend on Ω, the computation of $\Delta_{\Omega}^{t}+\Delta_{\Omega^C}^{t}$ mainly requires the summation over the shape Ω of the function $f^t(x,y)=f_A^t(x,y)-f_B^t(x,y)$, which can be performed in a very efficient way using integral images [2]. Indeed, the summation of $f^t(x,y)$ over the surface of any rectangle $\Omega=\{(x,y)\in[x_1,x_2]\times[y_1,y_2]\}$ can be obtained with four additions, provided the integral image $F^t(x,y)=\sum_{x_0\leq x,y_0\leq y} f^t(x_0,y_0)$ has been precomputed:

$$\sum_{(x,y)\in\Omega} f^t(x,y) = F^t(x_1,y_1)+F^t(x_2,y_2)-F^t(x_1,y_2)-F^t(x_2,y_1). \tag{9}$$

The minimization algorithm thus consists of first estimating the target translation and then in refining the target shape by deforming its rectangular contour as long as the stochastic complexity decreases. To reduce the computation time, the integral image at time t is calculated only on the search window, which corresponds to the target shape $\tilde{\Omega}^{t-1}$ estimated at time $t-1$ and dilated by 20 pixels. With such an implementation, the estimation of the target size and location can be performed in about 2 ms on 640 $\times$ 480 pixel images [with a standard 3.2 GHz personal computer using Linux and C programming].

This approach is illustrated on the video sequence of Fig. 1 (row 1), which has been acquired under snowy weather conditions. The rectangular shape obtained with the proposed method is displayed with continuous contours (see Fig. 1, row 1). These tracking results have been performed with using only the

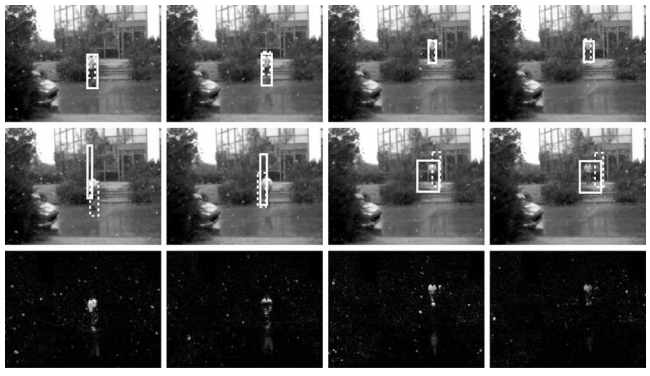

Fig. 1. Snowy video sequence (640 $\times$ 480 pixels). Comparison of the results obtained with the proposed stochastic complexity criterion (row 1) and the Bhattacharryya distance (row 2), when estimating both the target size and location (continuous contours) or only its location (dashed contours). Row 3, detection map obtained with a standard background subtraction method. Results obtained on the L^* component of the $L^*a^*b^*$ color-space (average computation time with the proposed method: 1.1 ms per frame).

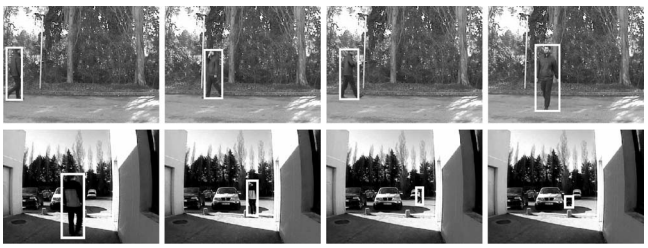

Fig. 2. Tracking results obtained on two outdoor video sequences (640×480 pixels). Row 1, results obtained on the L^* component of the $L^*a^*b^*$ color-space (2.0 ms per frame). Row 2, results obtained on the a^*b^* chromatic components of the $L^*a^*b^*$ color-space (1.9 ms per frame).

lightness information of the scene (the L^* component of the $L^*a^*b^*$ color space commonly employed in computer vision tasks as in [11]), i.e., only scalar images are used. As for all the results presented in this Letter, the target parameters m_A and σ_A^2 have been estimated the first time the target appears, and the background parameters $m_B(x,y)$ and $\sigma_B^2(x,y)$ have been learned by averaging the time series $\{s^t(x,y)\}_t$ before the target appears. To demonstrate the performance of the proposed stochastic complexity criterion, the results obtained when using a standard approach based on the Bhattacharryya distance (which consists of maximizing the similarity between the estimated Gaussian PDF inside Ω and the expected PDF P^A for the target) are shown in Fig. 1 (row 2), still with continuous contours. It clearly appears that the proposed method based on the stochastic complexity improves the results in comparison to the Bhattacharryya-based approach, since neither the target location nor its size have been correctly estimated in that latter case. It is thus shown in Fig. 1 that even if only the location of the target is estimated (the size of the shape is kept constant), the target is better located with the approach proposed in this Letter (dashed contours in row 1) than with the Bhattacharryya distance (dashed contours in row 2). Furthermore, the results of Fig. 1, row 3 demonstrate that contrary to the proposed approach, a standard background-subtraction method (based on the detection map $|s^t(x,y)-m_B(x,y)|/\sigma_B(x,y)$) is not sufficient to recover the target without strong regularization to detect the whole shape of the walking person and to reduce the high number of false alarms due notably to snowflakes.

The results obtained with the proposed approach on another video sequence are shown in Fig. 2 (row 1), still using the L^* component of the $L^*a^*b^*$ color space. Although this sequence presents a highly textured background with lightness quite similar to the target one, the target is correctly tracked all along the video sequence.

This approach can be generalized to vectorial images. For example, in Fig. 2 (row 2), the a^*b^* chromatic components of the $L^*a^*b^*$ color space are used. In this case, P^A and $P^B_{(x,y)}$ are assumed to be two-dimensional Gaussian PDFs with mean $\mathbf{m}_A$ and $\mathbf{m}_B(x,y)$ and covariance matrix $\mathbf{\Gamma}_A$ and $\mathbf{\Gamma}_B(x,y)$. Therefore, Eqs. (5) and (8) have to be generalized, leading to

$$\Delta_\Omega^t + \Delta_{\Omega^C}^t = K^t + N\log 2\pi + \frac{N_\Omega}{2}\log|\Gamma_A| + \frac{1}{2}\sum_{(x,y)\in\Omega}[f_A^t(x,y) - f_B^t(x,y)], \quad (10)$$

with

$$f_A^t(x,y) = [\mathbf{s}^t(x,y) - \mathbf{m}_A]^\dagger \times [\mathbf{\Gamma}_A]^{-1}[\mathbf{s}^t(x,y) - \mathbf{m}_A],$$

$$f_B^t(x,y) = \log|\mathbf{\Gamma}_B(x,y)| + [\mathbf{s}^t(x,y) - \mathbf{m}_B(x,y)]^\dagger[\mathbf{\Gamma}_B(x,y)]^{-1}[\mathbf{s}^t(x,y) - \mathbf{m}_B(x,y)], \quad (11)$$

where $\dagger$ denotes vector transposition and $|M|$ is the determinant of the matrix M. As shown in Fig. 2 (row 2), this generalization to vectorial images still leads to a reduced computation time. Moreover, these results demonstrate that this approach allows one to deal with strong variations of the target size without requiring the tuning of any parameter.

One of the perspectives of this Letter is to generalize it to a tracking framework for more complex tracking situations, such as multitargets, disappearing targets, and nonparametric PDFs while keeping very low computational costs.

The authors are grateful to Conseil Régional Provence-Alpes-Côte d'Azur for taking part in the funding of the Ph.D. of Jean-François Boulanger, and to Marc Allain for fruitful discussions.

References

1. J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, Vol. 15 of Series in Computer Science (World Scientific, 1989).
2. P. Viola and M. Jones, in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2000), p. 511.
3. C. Stauffer and W. E. L. Grimson, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 747 (2000).
4. Y. Sheikh and M. Shah, IEEE Trans. Pattern Anal. Mach. Intell. **27**, 1778 (2005).
5. L. Li, W. Huang, I. Yu-Hua Gu, and Q. Tian, IEEE Trans. Image Process. **13**, 1459 (2004).
6. M. Isard and A. Blake, International J. Comp. Vis. **29**, 5 (1998).
7. D. Comaniciu, V. Ramesh, and P. Meer, IEEE Trans. Pattern Anal. Mach. Intell. **25**, 564 (2003).
8. B. Zhang, W. Tian, and Z. Jin, Chin. Opt. Lett. **4**, 569 (2006).
9. R. Han, Z. Jing, and Y. Li, Chin. Opt. Lett. **6**, 168 (2008).
10. O. Ruch and P. Réfrégier, Opt. Lett. **26**, 977 (2001).
11. H.-L. Shen and J. H. Xin, Opt. Lett. **32**, 96 (2007).