We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,300
Open access books available

130,000
International authors and editors

155M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

2

Structural Information Approaches to Object Tracking in Video Sequences

Artur Loza[1], Lyudmila Mihaylova[2], Fanglin Wang[3] and Jie Yang[4]
[2] *Lancaster University*
[1,3,4] *Shanghai Jiao Tong University*
[2] *United Kingdom*
[1,3,4] *China*

1. Introduction

The problem of object tracking has been a subject of numerous studies and has gained considerable interest (Aghajan & Cavallaro (2009); Gandhi & Trivedi (2007)) in the light of surveillance (Hu et al. (2004); Loza et al. (2009)), pedestrian protection systems (Forsyth et al. (2006); Gerónimo et al. (2010); Smith & Singh (2006)), vehicular traffic management, human vision systems (Chen & Yang (2007)) and others. The methods for object tracking can be subdivided into two main groups: deterministic (Bradski (1998); Cheng (1995); Comaniciu & Meer (2002); Comaniciu et al. (2000; 2003)) and probabilistic (e.g., Pérez et al. (2004)) within which the Bayesian techniques are the most prominent.

Most video tracking techniques are region based which means that the object of interest is contained within a region, often of a rectangular or circular shape. This region is then tracked in a sequence of video frames based on certain features (or their histograms), such as colour, texture, edges, shape, and their combinations (Brasnett et al. (2005; 2007); Pérez et al. (2004); Triesch & von der Malsburg (2001)).

This book chapter addresses the problem of object tracking in video sequences by using the recently proposed structural similarity-based image distance measure (Wang et al., 2005a; Wang & Simoncelli, 2004). Advantages of the Structural SIMilarity (SSIM) measure are its robustness to illumination changes and ability to extract the structural information from images and video frames. Real world videos are often recorded in unfavourable environments, for example with low or variable light exposure due to the weather conditions. These factors often cause undesired luminance and contrast variations in videos produced by optical cameras (e.g. the object entering dark or shadowy areas) and by Infrared (IR) sensors (due to varying thermal conditions or insufficient exposure of the object). Moreover, due to the presence of spurious objects or backgrounds in the environment, real-world video data may lack sufficient colour information needed to discriminate the tracked object against its background.

The commonly applied tracking techniques relying on colour and edge image features represented by histograms are often prone to failure in such conditions. In contrast, the SSIM reflects the distance between two video frames by jointly comparing their luminance, contrast and spatial characteristics and is sensitive to relative rather than absolute changes in

the video frame. It replaces histograms used for calculation of the measurement likelihood function within a particle filter. We demonstrate that it is a good and efficient alternative to histogram-based tracking. This work builds upon the results reported in Loza et al. (2006; 2009) with more detailed investigation including further extensions of the proposed method.
The remaining part of the book chapter is organised in the following way. Section 2 presents an overview of the main (deterministic and probabilistic) tracking approaches and outlines the Bayesian tracking framework. Section 3 presents the proposed approach, followed in Section 4 by the results obtained with real video data and Section 5 summarises the results and open issues for future research.

2. Video tracking overview

2.1 Deterministic methods

In this chapter an overview of selected deterministic & probabilistic tracking techniques is presented. Within the group of deterministic methods the mean shift (MS) algorithm (Cheng (1995); Comaniciu & Meer (2002); Comaniciu et al. (2000; 2003)) is one of the most widely used. The MS algorithm originally proposed by Fukunaga & Hostetler (1975) was further extended to computer vision problems in (Comaniciu & Meer (2002); Comaniciu et al. (2000)). It is a gradient based, iterative technique that uses smooth kernels, such as Gaussian or Epanechnikov for representing a probability density function. The similarity between the target region and the target candidates in the next video frame is evaluated using a metric based on the Bhattacharyya coefficient (Aherne et al. (1990)). The MS tracking algorithm from Comaniciu & Meer (2002) is a mode-finding technique that locates the local maxima of the posterior density function. Based on the mean-shift vector, utilised as an estimate of the gradient of the Bhattacharyya function, the new object state estimate is calculated. The accuracy of the mean shift techniques depends on the kernel chosen and the number of iterations in the gradient estimation process. One of the drawbacks of the MS technique is that sometimes local extrema are found instead of the global one. Moreover, the MS algorithm faces problems with multimodal probability density functions which can be overcome by some of the Bayesian methods (sequential Monte Carlo methods).
The MS algorithm has been combined with particle filtering techniques and as a result kernel particle filters (Chang & Ansari (2003; 2005)) and hybrid particle filters (Maggio & Cavallaro (2005)) were proposed combining the advantages of both approaches. The MS is applied to the particles in order to move them into more likely regions and hence the performance of these hybrid particle filters is significantly improved. Interesting implementation of this scheme has been proposed in (Cai et al., 2006) where the data association problem is formulated and the MS algorithm is "embedded seamlessly" into the particle filter algorithm: the deterministic MS - induced particle bias with a superimposed Gaussian distribution is considered as a new proposal distribution. Other related hybrid particle filters combined with the MS have been proposed in (Bai & Liu (2007); Cai et al. (2006); Han et al. (2004); Shan et al. (2007)).

2.2 Bayesian tracking framework

Bayesian inference methods (Doucet et al. (2001); Isard & Blake (1998); Koch (2010); Pérez et al. (2004); Ristic et al. (2004)) have gained a strong reputation for tracking and data fusion applications, because they avoid simplifying assumptions that may degrade performance in complex situations and have the potential to provide an optimal or sub-optimal

solution (Arulampalam et al. (2002); Khan et al. (2005)). In case of the sub-optimal solution, the proximity to the theoretical optimum depends on the computational capability to execute numeric approximations and the feasibility of probabilistic models for target appearance, dynamics, and measurements likelihoods.
In the Bayesian tracking framework the best posterior estimate of the state vector $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is inferred from the available measurements, $\boldsymbol{z}_{1:k} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\}$, based on derivation of the posterior probability density function (pdf) of $\boldsymbol{x}_k$ conditioned on the whole set of observations: $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$. Assuming that the posterior pdf at time $k-1$ (the initial pdf) is available, the prior pdf of the state at time k is obtained via the Chapman-Kolmogorov equation:

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}) = \int_{\mathbb{R}^{nx}} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}) d\boldsymbol{x}_{k-1} \tag{1}$$

where $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is the state transition probability. Once the sequence $\boldsymbol{z}_{1:k}$ of measurements is available, the posterior pdf $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is recursively obtained according to the Bayes update rule

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \frac{p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1})}{p(\boldsymbol{z}_{1:k}|\boldsymbol{z}_{1:k-1})} \tag{2}$$

where $p(\boldsymbol{z}_{1:k}|\boldsymbol{z}_{1:k-1})$ is a normalising constant and $p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_k)$ is the measurement likelihood. Thus, the recursive update of $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is proportional to the measurement likelihood

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \propto p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}). \tag{3}$$

Different strategies can be applied to estimate $\boldsymbol{x}_k$ from this pdf. Commonly used estimators of $\boldsymbol{x}_k$, include the maximum a posteriori (MAP) approach,

$$\hat{\boldsymbol{x}}_k = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}), \tag{4}$$

and the minimum mean squared error (MMSE) approach, giving an estimate which is equivalent to the expected value of the state

$$\hat{\boldsymbol{x}}_k = \int \boldsymbol{x}_k p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) d\boldsymbol{x}_k. \tag{5}$$

2.3 Particle filtering techniques for state vector estimation

Particle filtering (Arulampalam et al. (2002); Doucet et al. (2001); Isard & Blake (1996; 1998); Pérez et al. (2004); Ristic et al. (2004)) is a method relying on sample-based reconstruction of probability density functions. The aim of sequential particle filtering is to evaluate the *posterior* pdf $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ of the state vector $\boldsymbol{x}_k$, given a set $\boldsymbol{z}_{1:k}$ of sensor measurements up to time k. The quality (importance) of the ℓ^{th} particle (sample) of the state, $\boldsymbol{x}_k^{(\ell)}$, is measured by the weight associated with it, $W_k^{(\ell)}$. An estimate of the variable of interest can be obtained by the weighted sum of particles (cf. (5) and (9)). The pseudo-code description of a generic particle filter (PF) tracking algorithm is shown in Table 1.
Two major stages can be distinguished in the particle filtering method: *prediction* and *update*. During prediction, each particle is modified according to the state model of the region of interest in the video frame, including the perturbation of the particle's state by means of addition of white noise in order to simulate the effect of the random walk according to the

Table 1. Pseudocode of the particle filter algorithm

Input: target state $\boldsymbol{x}_{k-1}$ (previous frame)
Output: target state $\boldsymbol{x}_k$ (current frame)

Initialisation
$k = 0$, initialise $\boldsymbol{x}_0$.
Generate N samples (particles) $\{\boldsymbol{x}_0^{(\ell)}\}, \ell = 1, 2, \ldots, N$, from the initial distribution $p(\boldsymbol{x}_0)$.
Initialise weights $W_0^{(\ell)} = 1/N$.

- FOR $k = 1 : K_{\text{frames}}$

* FOR $\ell = 1, 2, \ldots, N$

Prediction

1. Sample the state from the object motion model

$$\boldsymbol{x}_k^{(\ell)} \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(\ell)}). \tag{6}$$

Update

2. Evaluate the importance weights based on the likelihood $\mathcal{L}(\boldsymbol{z}_k | \boldsymbol{x}_k^{(\ell)})$ of the cue from the measurement $\boldsymbol{z}_k$

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)} \mathcal{L}(\boldsymbol{z}_k | \boldsymbol{x}_k^{(\ell)}). \tag{7}$$

* END FOR

Output

3. Normalise the weights of each particle

$$\widehat{W}_k^{(\ell)} = W_k^{(\ell)} / \sum_{\ell=1}^{N} W_k^{(\ell)}. \tag{8}$$

4. Compute the posterior mean state estimate of $\boldsymbol{x}_k$ using the collection of samples

$$\hat{\boldsymbol{x}}_k = \sum_{\ell=1}^{N} \widehat{W}_k^{(\ell)} \hat{\boldsymbol{x}}_k^{(\ell)}. \tag{9}$$

Resampling

5. Estimate the effective number of particles $N_{\text{eff}} = 1/\sum_{\ell=1}^{N} \left(\widehat{W}_k^{(\ell)}\right)^2$. If $N_{\text{eff}} \leq N_{\text{thr}}$ (N_{thr} is a given threshold) then perform resampling: multiply samples $\boldsymbol{x}_k^{(\ell)}$ with high importance weights $\widehat{W}_k^{(\ell)}$ and suppress samples with low importance weights, in order to introduce variety and obtain N new random samples. Set $W_k^{(\ell)} = \widehat{W}_k^{(\ell)} = 1/N$.

- END FOR

motion model $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(\ell)})$, (6). The prior pdf of the state at time k is obtained in prediction stage via Chapman-Kolmogorov equation (1). Once a measurement $\boldsymbol{z}_k$ is available, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$ is recursively obtained in the update step according to (3), or equivalently, (7). The likelihood $\mathcal{L}(\boldsymbol{z}_k|\boldsymbol{x}_k^{(\ell)})$ is calculated for the respective image cue (e.g. colour). Consequently, the posterior mean state is computed using the collection of particles (9).
An inherent problem with particle filters is degeneracy (the case when only one particle has a significant weight). A *resampling* procedure helps to avoid this by eliminating particles with small weights and replicating the particles with larger weights. Various approaches for resampling have been proposed (see Doucet et al. (2001); Kitagawa (1996); Liu & Chen (1998); Wan & van der Merwe (2001), for example). In this work, the systematic resampling method (Kitagawa (1996)) was used with the estimate of the measure of degeneracy (Doucet et al. (2001)) as given in (Liu & Chen (1998)) (see Table 1).

2.4 Importance sampling and proposal distributions

In the PF framework, the pdf of the object state, $p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k})$, is represented by a set of samples with associated weights $\{\boldsymbol{x}_k^{(\ell)}, W_k^{(\ell)}\}_{i=1}^N$ such that $\sum_{i=1}^N W_k^{(\ell)} = 1$. Then the posterior density can be approximated as

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \approx \sum_{\ell=1}^{N} W_k^{(\ell)} \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(\ell)}) \tag{10}$$

based on the likelihood $\mathcal{L}(\boldsymbol{z}_k|\boldsymbol{x}_k^{(\ell)})$ (see the following paragraph for details of the likelihood) of the measurement and particle weights. Here, $\delta(.)$ is the Dirac delta function. The particle weights in (10) are updated based on the principle of importance sampling (Arulampalam et al. (2002))

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)} \frac{p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(\ell)}) p(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)})}{q(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k})} , \tag{11}$$

where $q(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k})$ is a proposal, called an *importance density* and $p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(\ell)})$ is the measurement likelihood function. It has been assumed that $q(.)$ is only dependent on $\boldsymbol{x}_{k-1}^{(\ell)}$ and $\boldsymbol{z}_k$. The most popular choice of the importance density is the prior, $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$. This choice results in a simple implementation of the weight update stage (cf. (11))

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)} p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(\ell)}) . \tag{12}$$

However, using the transition information alone may not be sufficient to capture the complex dynamics of some targets. It has been shown that an optimal importance density is defined as function of the state and a new measurement/additional information $q(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k})$. Therefore, in this work, the use of a mixture distribution containing additional information as the importance density is proposed

$$q(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k}) = \sum_{m=1}^{M} \alpha_m f_m(\boldsymbol{x}_{1:k}^{(\ell)}, \boldsymbol{z}_{1:k}) \tag{13}$$

where $\alpha_m, \sum_{m=1}^M \alpha_m = 1$ are normalised weights of M components of the mixture. Among possible candidates for f_m are the prior, blob detection and data association distributions. For

$M = 1$ and $f_1(\boldsymbol{x}_{1:k}^{(\ell)}, \boldsymbol{z}_{1:k}) = p(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)})$ the generic PF is obtained. Examples of such mixture importance densities have been proposed in (Cai et al. (2006); Lu et al. (2009); Okuma et al. (2004)), consisting in an inclusion of the Adaboost detection information and a modification of the particle distribution with the use of a mode-seeking algorithm (M-S has been used). In this case the 'proposal distribution' has been defined as a mixture distribution between the prior and detection distributions:

$$q(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k}) = \alpha p_{\text{ada}}(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{z}_k) + (1-\alpha)p(\boldsymbol{x}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}) \tag{14}$$

In (Cai et al., 2006) the application of M-S optimisation to the particles is considered as a new proposal distribution:

$$\breve{q}(\breve{\boldsymbol{x}}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k}) = \mathcal{N}(\breve{\boldsymbol{x}}_k^{(\ell)}|\tilde{\boldsymbol{x}}_k^{(\ell)}, \boldsymbol{\Sigma}) \,, \tag{15}$$

where $\tilde{\boldsymbol{x}}_k^{(\ell)}$ are M-S-modified samples of the original proposal distribution q and $\mathcal{N}(\breve{\boldsymbol{x}}_k^{(\ell)}|\tilde{\boldsymbol{x}}_k^{(\ell)}, \boldsymbol{\Sigma})$ is a Gaussian distribution, with mean $\tilde{\boldsymbol{x}}_k^{(\ell)}$ fixed covariance $\boldsymbol{\Sigma}$, superimposed on the results of M-S. The particle weights are then updated accordingly, i.e.

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)} \frac{p(\boldsymbol{z}_k|\breve{\boldsymbol{x}}_k^{(\ell)})p(\breve{\boldsymbol{x}}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)})}{q(\breve{\boldsymbol{x}}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \boldsymbol{z}_{1:k})} \,. \tag{16}$$

3. The structural information approach

The recently proposed approach combining the SSIM and particle filtering for video tracking has been shown in (Loza et al., 2009) to outperform similar methods using the conventional colour or edge histograms and Bhattacharyya distance. However, the structural similarity combined with the particle filtering approach results in increased computational complexity of the algorithm due to the necessity of extracting the structural information at each point of the state space. In this book chapter, novel optimised approaches based on the SSIM are proposed for video tracking. Firstly, a fast, deterministic version of the SSIM-based tracking algorithm is developed. The deterministic tracking algorithm estimates the state of the target (location and size) combining a gradient ascent procedure with the structural similarity surface of the current video frame, thus avoiding computationally expensive sampling of the state space. Next, an optimisation scheme is presented, based on a hybrid PF with a deterministic mode search, applied to the particle distribution.

3.1 Structural similarity measure

The proposed method uses a similarity measure computed directly in the image spatial domain. This approach differs significantly from other particle filtering algorithms, that compare image distributions represented by their sample histograms (Nummiaro et al. (2003); Pérez et al. (2004); Shen et al. (2003)).

Although many simple image similarity measures exist (for example, mean square error, mean absolute error or peak signal-to-noise ratio), most of these have failed so far to capture the perceptual similarity of images/video frames under the conditions of varied luminance, contrast, compression or noise (Wang et al. (2004)). Recently, based on the premise that the HVS is highly tuned to extracting structural information, a new image metric has been developed, called the Structural SIMilarity (SSIM) index (Wang et al. (2004)). The SSIM index,

between two images, I and J is defined as follows:

$$S(I,J) = \left(\frac{2\mu_I\mu_J + C_1}{\mu_I^2 + \mu_J^2 + C_1}\right)\left(\frac{2\sigma_I\sigma_J + C_2}{\sigma_I^2 + \sigma_J^2 + C_2}\right)\left(\frac{\sigma_{IJ} + C_3}{\sigma_I\sigma_J + C_3}\right) \tag{17}$$

$$= l(I,J)\, c(I,J)\, s(I,J),$$

where $C_{1,2,3}$ are small positive constants used for the numerical stability purposes, μ denotes the sample mean

$$\mu_I = \frac{1}{L}\sum_{j=1}^{L} I_j, \tag{18}$$

σ denotes the sample standard deviation

$$\sigma_I = \sqrt{\frac{1}{L-1}\sum_{j=1}^{L}(I_j - \mu_I)^2} \tag{19}$$

and

$$\sigma_{IJ} = \frac{1}{L-1}\sum_{j=1}^{L}(I_j - \mu_I)(J_j - \mu_J) \tag{20}$$

corresponds to the sample covariance. The estimators are defined identically for images I and J, each having L pixels. The image statistics are computed in the way proposed in (Wang et al. (2004)), i.e. locally, within a 11×11 normalised circular-symmetric Gaussian window.
For $C_3 = C_2/2$, (17) can be simplified to obtain

$$S(I,J) = \left(\frac{2\mu_I\mu_J + C_1}{\mu_I^2 + \mu_J^2 + C_1}\right)\left(\frac{2\sigma_{IJ} + C_2}{\sigma_I^2 + \sigma_J^2 + C_2}\right). \tag{21}$$

3.2 Selected properties of the SSIM

The three components of (17), l, c and s, measure respectively the *luminance*, *contrast* and *structural similarity* of the two images. Such a combination of image properties can be seen as a fusion of three independent image cues. The relative independence assumption is based on a claim that a moderate luminance and/or contrast variation does not affect structures of the image objects (Wang et al. (2005a)).
In the context of the multimodal data used in our investigation, an important feature of the SSIM index is (approximate) invariance to certain image distortions. It has been shown in (Wang et al. (2005a; 2004)), that the normalised luminance measurement, l, is sensitive to the relative rather than to absolute luminance change, thus following the masking feature of the Hue, Saturation, Value (HVS).
Similarly, the contrast comparison function, c, is less sensitive to contrast changes occurring in images with high base contrast. Finally, the structure comparison, s, is performed on contrast-normalised signal with mean luminance extracted, making it immune to other (non-structural) distortions.
These particular invariance properties of the SSIM index make it suitable for the use with multimodal and surveillance video sequences. The similarity measure is less sensitive to

the type of global luminance and contrast changes produced by infrared sensors (results of varied thermal conditions or exposure of the object) and visible sensors (for example, the object entering dark or shadowy areas or operating in variable lighting conditions). Moreover, the structure comparison is expected to be more reliable in scenarios when spurious objects appear in the scene or when there is not enough discriminative colour information available. The latter may be the result of the tracked object being set against background of similar colour or when background-like camouflage is deliberately being used.
It can easily be shown that the measure defined in (17) is symmetric, i.e.

$$S(I,J) = S(J,I) \tag{22}$$

and has a unique upper bound

$$S(I,J) \leq 1,\ S(I,J) = 1 \text{ iff } I = J. \tag{23}$$

One way of converting such a similarity $S(I,J)$ into dissimilarity $D(I,J)$ is to take (Loza et al. (2009); Webb (2003))

$$D(I,J) = \frac{1}{|S(I,J)|} - 1. \tag{24}$$

Here a more natural way Webb (2003),

$$D(I,J) = (1 - S(I,J))/2. \tag{25}$$

is preferred, however, as it maps the dissimilarity into [0, 1] interval (0 when the images are identical). The measure (25) satisfies non-negativity, reflexivity and symmetry conditions. Although sufficient for our purposes, this dissimilarity measure is not a metric, as it does not satisfy the triangle condition. In the following Section we present a method of evaluating the likelihood function, based on the structural similarity between two greyscale images.

3.3 The structural information particle filter tracking algorithm

Below the main constituents of the structural similarity-based particle filter tracking algorithm (SSIM-PF), such us motion, likelihood and target model, are described. A pseudocode of the algorithm is shown in Table 2.

3.3.1 Motion model

The motion of the moving object can be modelled by the random walk model,

$$\boldsymbol{x}_k = \boldsymbol{F}\boldsymbol{x}_{k-1} + \boldsymbol{v}_{k-1}, \tag{26}$$

with a state vector $\boldsymbol{x}_k = (x_k, y_k, s_k)^T$ comprising the pixel coordinates (x_k, y_k) of the centre of the region surrounding the object and the region scale s_k; $\boldsymbol{F}$ is the transition matrix ($\boldsymbol{F} = \boldsymbol{I}$ in the random walk model) and $\boldsymbol{v}_k$ is the process noise assumed to be white, Gaussian, with a covariance matrix

$$\boldsymbol{Q} = \text{diag}(\sigma_x^2, \sigma_y^2, {\sigma_s}^2). \tag{27}$$

The estimation of the scale permits adjustment of the region size of the moving objects, e.g., when it goes away from the camera, when it gets closer to it, or when the camera zoom varies. Depending on the type of the tracking object and the environment in which tracking is

Table 2. Pseudocode of the SSIM-based particle filter algorithm

Input: target state $\boldsymbol{x}_{k-1}$ (previous frame)
Output: target state $\boldsymbol{x}_k$ (current frame)

Initialisation
$k = 0$, initialise tracked region at $\mathbf{x}_0$.
Generate N samples (particles) $\{\boldsymbol{x}_0^{(\ell)}\}, \ell = 1, 2, \ldots, N$, from the initial distribution $p(\boldsymbol{x}_0)$.
Initialise weights $W_0^{(\ell)} = 1/N$.
• FOR $k = 1 : K_{\text{frames}}$
* FOR $\ell = 1, 2, \ldots, N$

Prediction

1. Sample the state from the object motion model $\boldsymbol{x}_k^{(\ell)} \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(\ell)})$.

Update

3. Evaluate the importance weights according to 29:

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)} \mathcal{L}(\boldsymbol{z}_k | \boldsymbol{x}_k^{(\ell)}). \tag{28}$$

* END FOR

Output

4. Normalise the weights of each particle (8)
5. Compute the posterior mean state estimate of $\boldsymbol{x}_k$ (9).

Resampling

6. Perform resampling as described in Table 1

• END FOR

performed, the state vector can be extended to include, for example, the acceleration variables, and the fixed ratio condition can be relaxed allowing independent changes of the height and the width of the object. However, increased dimensionality of the state vector requires finer sampling of the state space, and thus undesirably high number of particles, which may preclude real-time implementation of the tracking system.

3.3.2 Likelihood model

The distance between the reference (target) region $\boldsymbol{t}_{ref}$ and the current region $\boldsymbol{t}_{\boldsymbol{k}}$ is calculated by the similarity measure (25). The normalised distance between the two regions is then substituted into the likelihood function, modelled as an exponential:

$$\mathcal{L}(\boldsymbol{z}_k | \boldsymbol{x}_k^{(\ell)}) \propto \exp\left(-D^2(\boldsymbol{t}_{\text{ref}}, \boldsymbol{t}_k)/D_{\text{min}}^2\right), \tag{29}$$

where $D_{\min} = \min_{\boldsymbol{x}}\{D(\boldsymbol{t}_{\text{ref}}, \boldsymbol{t}_k)\}$. Here $\boldsymbol{z}$ denotes the measurement vector, although with the SSIM a measurement in explicit form is not available. This smooth likelihood function, although chosen empirically by (Pérez et al. (2004)), has been in widespread use for a variety of cues ever since. The similarity-based distance proposed in this work is an alternative to the Bhattacharyya distance D, commonly used to calculate similarity between target and reference objects, described by their histograms h:

$$D(\boldsymbol{t}_{\text{ref}}, \boldsymbol{t}_k) = \left(1 - \sum_{i=1}^{B} h_{\text{ref},i} h_{k,i}\right)^{0.5}. \tag{30}$$

where the tracked image regions are described by their colour (Nummiaro et al. (2003)) or texture histograms (Brasnett et al. (2007)). The likelihood function is then used to evaluate the importance weights of the particle filter, to update the particles and to obtain the overall estimate of the centre of the current region.

3.3.3 Target model

The tracked objects are defined as image regions within a rectangle or ellipsoid specified by the state vector (i.e. spatial location and scale). In the particle filtering framework as specified in Table 1, a region corresponding to each particle, centred at location (x, y) and resized according to the scale parameter of the state, is computed. The extracted region is then compared to the target region using the distance measure D (25). The structural properties of the region extracted through SSIM (17) are related with the estimates of the centre of the region of interest and are used directly to calculate the distance D in (29) between the reference and current region as shown in (25).

3.4 Differential SSIM tracking algorithm

In the SSIM-PF tracking algorithm, described in Section 3.3, the SSIM is computed a large number of times, i.e. for each particle. This makes the SSIM-PF method computational expensive when a large number of particles is required. In this section, a low-complexity, deterministic alternative to the SSIM-PF is proposed, namely Differential SSIM-based tracker (DSSIM). The proposed algorithm tracks the object by analysing the gradient SSIM surface computed between the current video frame and the object model. This deterministic iterative gradient search procedure uses the structural information directly and does not rely on the probabilistic framework introduced in Section 2.2.

In order to achieve a computationally efficient tracking performance, whilst retaining the benefits of the original measure, a differential SSIM formula is proposed as follows. The object is tracked in the spatial domain of the subsequent video frames by maximising the measure (21) with respect to location $\mathbf{x}$, based on its gradient. In order to simplify the subsequent derivation, we choose to analyse the logarithm of (21) by defining a function $\rho(\boldsymbol{x})$:

$$\rho(\boldsymbol{x}) = s \log(|S(\boldsymbol{x})|) \tag{31}$$

$$= s \log\left(2\mu_I\mu_J + C_1\right) - \log\left(\mu_I^2 + \mu_J^2 + C_1\right) + \log\left(2|\sigma_{IJ}| + C_2\right) - \log\left(\sigma_I^2 + \sigma_J^2 + C_2\right). \tag{32}$$

Table 3. Pseudocode of the proposed DSSIM tracking algorithm

Input: target state $\boldsymbol{x}_{k-1}$ (previous frame)
Output: target state $\boldsymbol{x}_k$ (current frame)

Initialisation

$k = 0$, initialise tracked region at $\boldsymbol{x}_0$.

- FOR $k = 1 : K_{\text{frames}}$

0. Initialise $\boldsymbol{x}_k^{(0)} = \boldsymbol{x}_k^{(1)} = \boldsymbol{x}_{k-1}$

* WHILE $S(\boldsymbol{x}_k^{(1)}) \geq S(\boldsymbol{x}_k^{(0)})$

1. Assign $\boldsymbol{x}_k^{(0)} = \boldsymbol{x}_k^{(1)}$
2. Calculate $\nabla\rho(\boldsymbol{x}_k^{(0)})$ according to (39)
3. Assign $\boldsymbol{x}_k^{(1)}$ the location of a pixel in $\boldsymbol{x}_k^{(0)}$ 8-connected neighbourhood, along the direction of $\nabla\rho(\boldsymbol{x}_k^{(0)})$

* END WHILE

Output

4. Assign target location in the current frame $\boldsymbol{x}_k = \boldsymbol{x}_k^{(0)}$

- END FOR

where $S(\boldsymbol{x})$ denotes the similarity (21) between the object template J and a current frame image region I centered around the pixel location $\boldsymbol{x} = (x, y)$ and $s = \text{sign}\,(S(\boldsymbol{x}))$. After a simple expansion of (31) we obtain the expression for the gradient of the function $\rho(\boldsymbol{x})$

$$\nabla\rho(\boldsymbol{x}) = s\left(A_1\nabla\mu_I + A_2\nabla\sigma_I^2 + A_3\nabla\sigma_{IJ}\right)\,, \tag{33}$$

where

$$A_1 = \frac{2\mu_J}{2\mu_I\mu_J + C_1} - \frac{2\mu_I}{\mu_I^2 + \mu_J^2 + C_1}\,, \tag{34}$$

$$A_2 = -\frac{1}{\sigma_I^2 + \sigma_J^2 + C_2}\,, \quad A_3 = \frac{1}{2\sigma_{IJ} + C_2}\,. \tag{35}$$

The gradients $\nabla\mu_I$ and $\nabla\sigma_I^2$ can be calculated as follows

$$\nabla\mu_I = \frac{1}{L}\sum_{i=1}^{L}\nabla I_i\,, \tag{36}$$

$$\nabla\sigma_I^2 = \frac{2}{L-1}\sum_{i=1}^{L}(I_i - \mu_I)\nabla I_i\,. \tag{37}$$

A simplified expression for the covariance gradient, $\nabla\sigma_{IJ}$, can be obtained, based on the observation that $\sum_{i=1}^{L}(J_i - \mu_J) = 0$:

$$\begin{aligned}\nabla\sigma_{IJ} &= \frac{1}{L-1}\sum_{i=1}^{L}(J_i - \mu_J)(\nabla I_i - \nabla\mu_I) \\ &= \frac{1}{L-1}\sum_{i=1}^{L}(J_i - \mu_J)\nabla I_i\end{aligned} \tag{38}$$

Finally, by defining the gradient of the pixel intensity as $\nabla I_i = \left(\frac{\partial I_i}{\partial x}, \frac{\partial I_i}{\partial y}\right)^T$, the complete formula for $\nabla\rho(\boldsymbol{x})$ is obtained

$$\nabla\rho(\boldsymbol{x}) = s\sum_{i=1}^{L}\left(\frac{A_1}{L} + \frac{2A_2(I_i - \mu_I) + A_3(J_i - \mu_J)}{L-1}\right)\nabla I_i\,. \tag{39}$$

The proposed algorithm, employing the gradient DSSIM function (39) is summarised in Table 3. In general terms, the estimated target location, $\mathbf{x}_k^{(0)}$ is moved along the direction of the structural similarity gradient by one pixel in each iteration until no further improvement is achieved. The number of SSIM and gradient evaluations depends on the number of iterations needed to find the maximum of the measure $S(\mathbf{x})$ and on average does not exceed 5 in our experiments. This makes our approach significantly faster than the original SSIM-PF. It should be noted that although the differential framework of the algorithm is based on a reformulation of the scheme proposed in (Zhao et al. (2007)), it utilises a distinct similarity measure.

3.5 The hybrid SSIM-PF tracker algorithm

An extension to the SSIM-PF, by deterministically modifying each particle according to the local structural similarity surface, referred to as hybrid SSIM-PF, is proposed in this correspondence. In the DSSIM procedure described in Section 3.4, the estimated target location, $\boldsymbol{x}_k^{(0)}$ is moved along the direction the structural similarity gradient by one pixel in each iteration until no further improvement is achieved, or the limit of iterations is reached. In the hybrid scheme proposed here this step is performed for each particle, following its prediction (step 1. in Table 1). In accordance with the principle of importance sampling (see Section 2.4), the prior distribution p resulting from the particle prediction and the proposal distribution q centred on the optimised position of the particle in the state space, are used to re-calculate the weight of a resulting particle $\tilde{\boldsymbol{x}}^{(\ell)}$:

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)}\frac{\mathcal{L}(\boldsymbol{z}_k|\tilde{\boldsymbol{x}}_k^{(\ell)})\,p(\tilde{\boldsymbol{x}}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)})}{q(\tilde{\boldsymbol{x}}_k^{(\ell)}|\boldsymbol{x}_{k-1}^{(\ell)}, \mathbf{z}_k)}\,. \tag{40}$$

with the proposal distribution defined analogously to Lu et al. (2009)

$$q(\tilde{\boldsymbol{x}}_k^{(\ell)}|\tilde{\boldsymbol{x}}_{k-1}^{(\ell)}, \boldsymbol{z}_k) = \alpha p_{\mathrm{DSSIM}}(\tilde{\boldsymbol{x}}_k^{(\ell)}|\boldsymbol{z}_k) + (1-\alpha)p(\tilde{\boldsymbol{x}}_k^{(\ell)}|\tilde{\boldsymbol{x}}_{k-1}^{(\ell)})\,. \tag{41}$$

In our implementation of this algorithm the mixing parameter is set to $\alpha = 0.5$ resulting in a uniform mixture distribution of two Gaussian distributions with identical covariances (27),

Table 4. Pseudocode of the hybrid particle filter algorithm

Input: target state $\boldsymbol{x}_{k-1}$ (previous frame)
Output: target state $\boldsymbol{x}_k$ (current frame)

Initialisation
$k = 0$, initialise tracked region at $\boldsymbol{x}_0$.
Generate N samples (particles) $\{\boldsymbol{x}_0^{(\ell)}\}, \ell = 1, 2, \ldots, N$, from the initial distribution $p(\boldsymbol{x}_0)$.
Initialise weights $W_0^{(\ell)} = 1/N$.
• FOR $k = 1 : K_{\text{frames}}$
* FOR $\ell = 1, 2, \ldots, N$

Prediction

1. Sample the state from the object motion model $\boldsymbol{x}_k^{(\ell)} \sim p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}^{(\ell)})$.

Optimisation

2. Modify the particle associated with the state $\boldsymbol{x}_k^{(\ell)}$ by performing steps 0.–4., Table 3.

   Assign the modified state to $\tilde{\boldsymbol{x}}_k^{(\ell)}$.

Update

3. Evaluate the importance weights

$$W_k^{(\ell)} \propto W_{k-1}^{(\ell)} \frac{\mathcal{L}(\boldsymbol{z}_k | \tilde{\boldsymbol{x}}_k^{(\ell)}) p(\tilde{\boldsymbol{x}}_k^{(\ell)} | \boldsymbol{x}_{k-1}^{(\ell)})}{q(\tilde{\boldsymbol{x}}_k^{(\ell)} | \boldsymbol{x}_{k-1}^{(\ell)}, \mathbf{z}_k)} \,. \tag{42}$$

   with proposal distribution q defined as in (41).

* END FOR

Output

4. Normalise the weights of each particle (8)
5. Compute the posterior mean state estimate of $\boldsymbol{x}_k$ (9).

Resampling

6. Perform resampling as described in Table 1

• END FOR

centred on the motion model-predicted particle and its optimised version, respectively. The proposed method is described in the form of a pseudocode in Table 4.

Fig. 1. Reference frames from the test videos

4. Tracking performance

4.1 Evaluation metrics

Tracking algorithms are usually evaluated based on whether they generate correct mobile object trajectories. In addition to the commonly applied visual assessment of the tracking performance, qualitative measures can be used to provide formal comparisons of the tested algorithms. In our work, the Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(k) = \left(\frac{1}{M} \sum_{m=1}^{M} (x_k - \hat{x}_{k,m})^2 + (y_k - \hat{y}_{k,m})^2 \right)^{\frac{1}{2}} \tag{43}$$

has been used as numerical measure of the performance of the developed techniques. In (43) $(\hat{x}_{k,m}, \hat{y}_{k,m})$ stand for the upper-left corner coordinates of the tracking box determined by both the object's central position and the scale estimated by the tracking algorithm in the frame k in m-th independent simulation (in our simulations $M = 50$ for probabilistic tracking algorithms and $M = 1$ for DSSIM and MS). The corresponding ground truth positions of the object, (x_k, y_k), have been generated by manually tracking the object.

4.2 Video sequences

The performance of our method is demonstrated over various multimodal video sequences, in which we aim to track a pre-selected moving person. The sequence *cross* (5 sec duration), taken from our multimodal database *The Eden Project Multi-Sensor Data Set* (2006), contains three people walking rapidly in front of a stationary camera. The main difficulties posed by this sequence are: the colour similarity between the tracked object and the background or other passing people, and a temporal near-complete occlusion of the tracked person by a passer-by.

Seq. name	mean RMSE				std RMSE			
	colour	edges	col.&edges	SSIM	colour	edges	col.&edges	SSIM
cross	150.5	77.4	39.6	8.3	98.4	70.1	58.2	5.1
man	71.5	27.7	48.4	8.0	46.1	23.7	34.3	6.5
bushes_ir	71.9	30.7	26.9	21.0	40.8	9.6	7.8	7.5
bushes_vi	98.4	36.0	36.4	19.1	13.1	15.7	16.7	7.1
bushes_cwt	92.6	45.4	32.0	20.7	54.6	21.0	12.1	7.5

Table 5. The performance evaluation measures of the tracking simulations

The sequence *man* (40 sec long), has been obtained from PerceptiVU, Inc. (n.d.). This is a video showing a person walking along a car park. Apart from the object's similarity to the nearby cars and the shadowed areas, the video contains numerous instabilities. These result from a shaking camera (changes in the camera pan and tilt), fast zoom-ins and zoom-outs, and a altered view angle towards the end of the sequence.
The three multimodal sequences *bushes* (*The Eden Project Multi-Sensor Data Set* (2006)), contain simultaneous registered infrared (*ir*), visual (*vi*) and complex wavelet transform fused (*cwt*, see Lewis et al. (2007) for details) recordings of two camouflaged people walking in front of a stationary camera (10 sec). The tracked individual looks very similar to the background. The video contains changes in the illumination (the object entering shadowy areas) together with nonstationary surroundings (bushes moved by strong wind). The reference frames used in tracking are shown in Figure 1.

4.3 Comparison of tracking cues

In this section the commonly used tracking cues (colour, edge histograms and their combination (Brasnett et al. (2007); Nummiaro et al. (2003))) are compared with the cue based on the structural similarity information. In order to facilitate easy and fair comparison the cues are evaluated in the same PF framework with identical initialisation and common parameters. The reference targets shown in Figure 1 were tracked in 50 Monte Carlo simulations and then the tracking output of each cue has been compared to the ground truth. The exemplary frames showing the tracking output are given in Figures 2–4 and the mean of RMSE and its standard deviation (std) were computed and are presented in Table 5.
From inspection of the video output in Figures 2–4 and the tracking error statistics in Table 5 it can clearly be seen that the SSIM-based method outperforms the other methods in all instances while never loosing the tracked object. The colour-based PF algorithm is the most prone to fail or give imprecise estimates of the object's state. Combining edge and colour cues is usually beneficial, however in some cases (sequences *man* and *bushes_vi*) the errors of the colour-based PF propagate through the performance of the algorithm, making it less precise than the PF based on edges alone. Another observation is that the 'structure' tracking algorithm has been least affected by the modality of *bushes* and the fusion process, which demonstrates the robustness of the proposed method to luminance and contrast alterations.
A closer investigation of the selected output frames illustrates the specific performance of the different methods. Figures 2–4 show the object tracking boxes constructed from the mean locations and scales estimated during the tests. Additionally, the particles and object location obtained from one of the Monte Carlo trials are shown. Since very similar performance has been obtained for all three *bushes* videos, only the fused sequence, containing complementary information from both input modalities, is shown. The visual difference between contents of

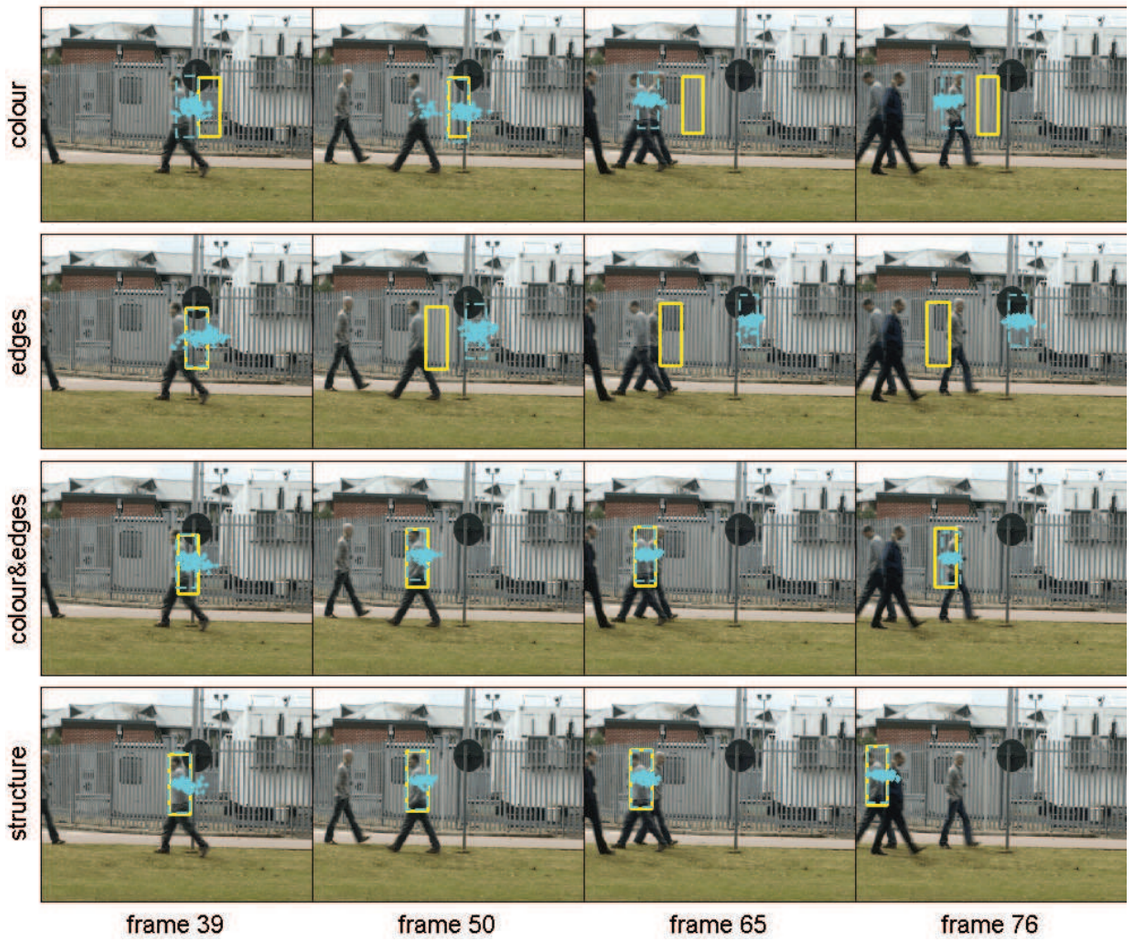


Fig. 2. Example video frames with average output of the tracking algorithm (solid line rectangle), a single trial output (dashed line rectangle and particles) superimposed, sequence *cross*

the input *bushes* videos (colour information, a person hidden in shaded area) can be seen by comparing the reference frames in Figure 1.

In the sequence *cross*, Figure 2, the 'colour' and 'edges' tracking algorithms are distracted by the road sign, which eventually leads to the loss of the object. Then, the first non-occluding passer-by causes the 'colour&edges' cue tracking algorithm to loose the object (frame 65). The 'structure' tracking technique is not distracted even by the temporary occlusion (frame 76).

The shaking camera in the sequence *man* (Figure 3, frame 162), has less effect on the 'structure' tracking technique than on the other compared algorithms, which appear to choose the wrong scale of the tracking box. Moreover, the other considered tracking algorithms do not perform well in case of similar dark objects appearing close-by (shadows, tyres, frame 478, where the 'colour' tracking algorithm permanently looses object) and rapid zoom-in (frame 711) and zoom-out of the camera (frame 790). Our method, however, seems to cope well with both situations. It should be noted, however, that 'colour&edges' (and 'edges') based algorithms show a good ability of recovering from some of the failings.

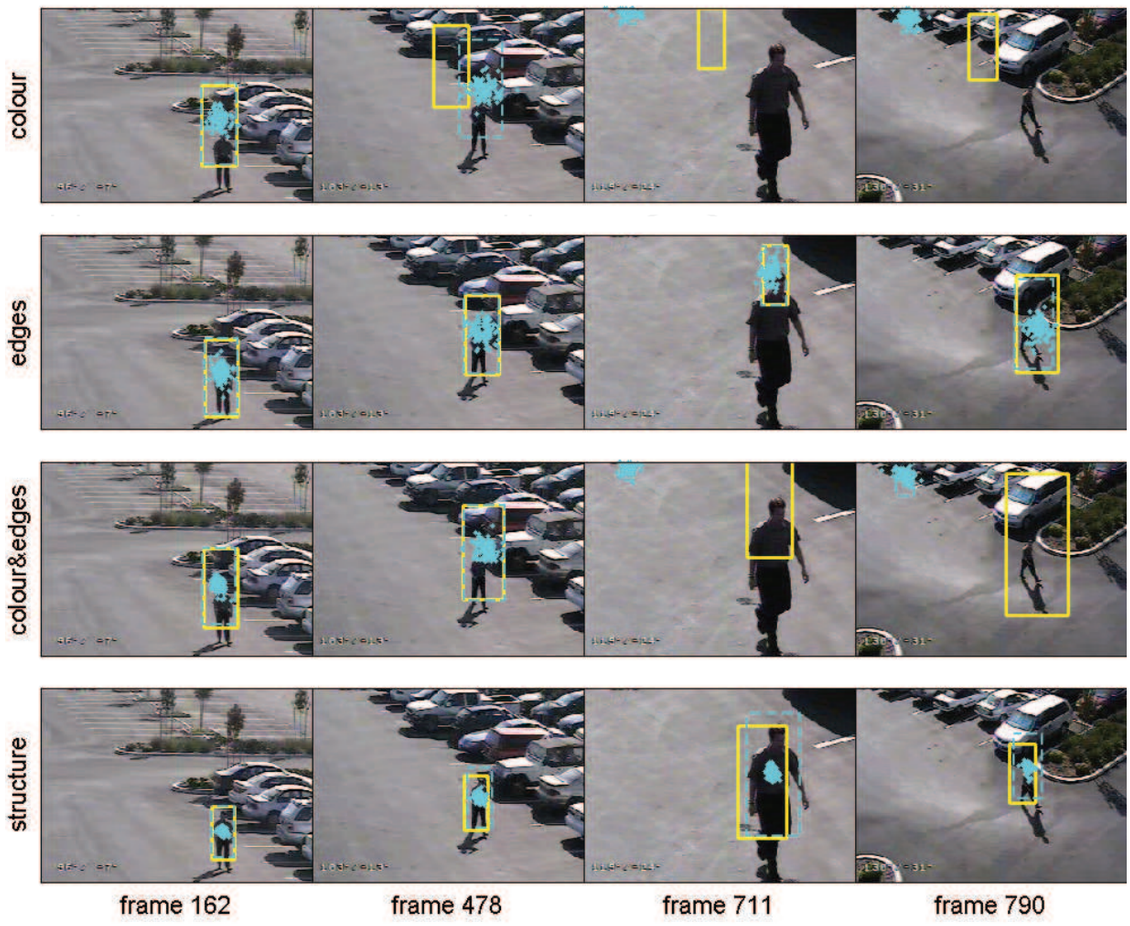


Fig. 3. Example video frames with average output of the tracking algorithm (solid line rectangle), a single trial output (dashed line rectangle and particles) superimposed, sequence *man*

Similarly, in the multimodal sequence *bushes*, Figure 4, the proposed 'structure' tracking algorithm is the most precise and the 'colour' tracking algorithm the least precise. The use of the fused video, although resulting in slightly deteriorated performance of the 'edges' based tracking algorithm, can still be motivated by the fact that it retains complementary information useful both for the tracking algorithm and a human operator (Cvejic et al. (2007); Mihaylova et al. (2006)): contextual information from the visible sequence and a hidden object location from the infrared sequence.

A single-trial output shown in Figures 2–4 exemplifies the spread of the spatial distribution of the particles. Typically, in the 'structure' tracking technique, particles are the most concentrated. Similar features can be observed in the output of the 'colour&edges' tracking algorithm. The particle distribution of the remaining PF tracking algorithms is much more spread, often attracted by spurious objects (see Figures 2 and 3, in particular).

It should also be noted that, the tracking performance varies between realisations, often giving different results compared with the output averaged over all Monte Carlo trials. Also in this

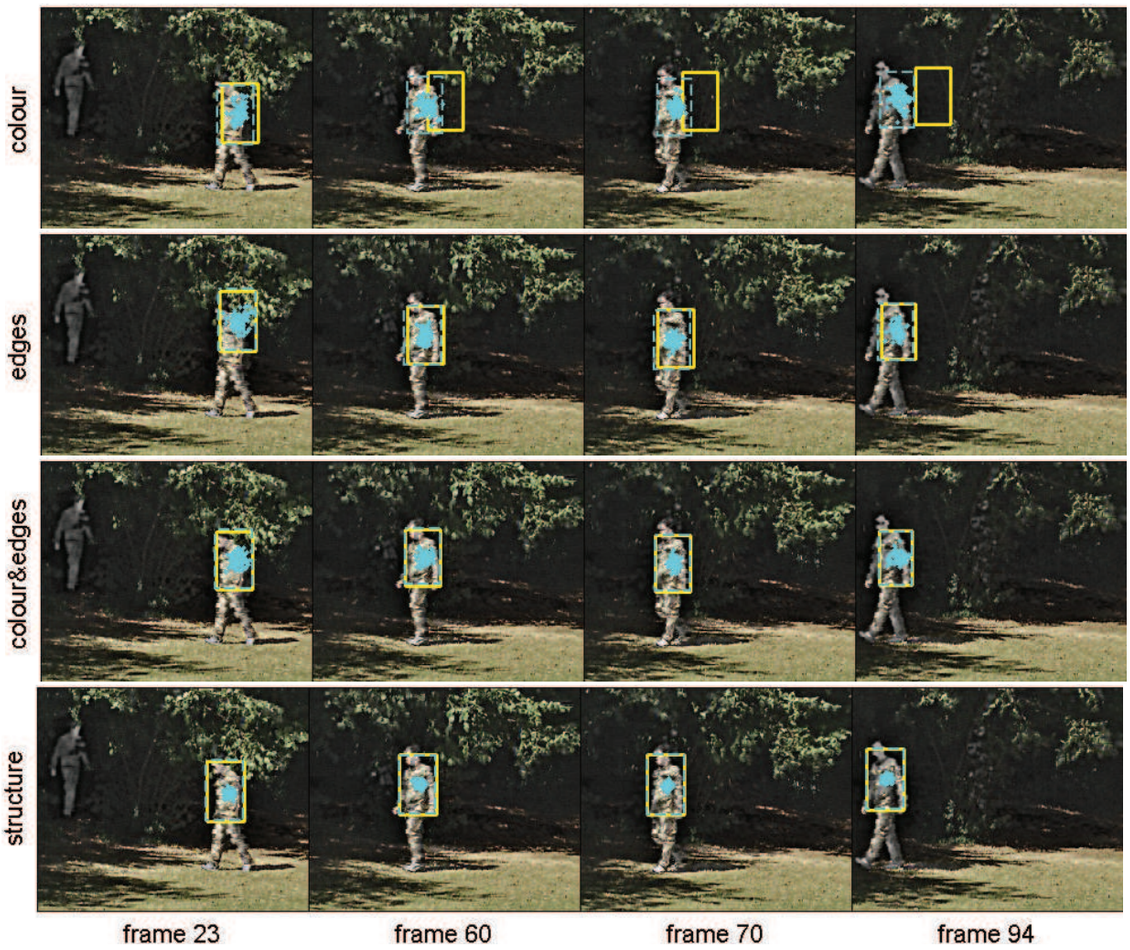


Fig. 4. Example video frames with average output of the tracking algorithm (solid line rectangle), a single trial output (dashed line rectangle and particles) superimposed, sequence *bushes_cwt*

respect the proposed method has been observed to be the most consistent, i.e., its results had the lowest variation, as illustrated by the low values of std RMSE in Table 5.

4.4 Comparison of SSIM-based probabilistic and deterministic techniques

In this section the probabilistic tracking SSIM-PF (Section 3.3) is compared with its deterministic counterpart, DSSIM-PF (Section 3.4). Since the main motivation for development of DSSIM technique was the reduction of the computational complexity, the algorithms are also evaluated with respect to their execution speed and therefore the rate at which the video frames are processed by the tracking algorithms, measured as frames per second (FPS), has been included in the results shown in Table 6. The proposed algorithms have been compared with another deterministic technique, the MS algorithm (Comaniciu & Meer (2002)). Analogously to PF-based methods, the MS and DSSIM algorithms are made scale-adaptive, by varying the object size by 5% and choosing the size giving the best match in terms of the similarity measure used.

Seq. name	Image size (pixels)	Speed (fps)			Mean RMSE (pixels)			std RMSE (pixels)		
		MS	SSIM-PF	DSSIM	MS	SSIM-PF	DSSIM	MS	SSIM-PF	DSSIM
cross	720×576	27	13	71	37.3	8.3	5.6	62.2	5.1	4.3
man	320×240	109	53	315	18.2	8.0	7.0	13.1	6.5	4.9

Table 6. The performance evaluation measures of the tracking simulations

Based on the performance measures in Table 5, it can be concluded that DSSIM outperforms the MS and SSIM-PF, both in terms of the processing speed and the tracking accuracy. It also appears to be more stable than the other two methods (lowest std). Although the example frames in Figure 5 reveal that in a number of instances the methods perform comparably, it can be seen that DSSIM method achieves the overall best performance in most of the frames. Admittedly, the difference between the accuracy and the stability of SSIM-PF and DSSIM is not large in most cases, however, in terms of the computational complexity, DSSIM method compares much more favourably with the other two techniques. The average tracking speed estimates were computed on PC in the following setup: CPU clock 2.66 GHZ, 1G RAM, MS and DSSIM requiring on average 20 and 5 iterations, respectively, and PF using 100 particles. In terms of the relative computational efficiency, the proposed method has been found to be approximately four times faster than SSIM-PF and twice as fast as MS.
The exemplary frames in Figure 5, where the 'difficult' frames have been selected, offer more insight into the performance and robustness of the algorithms. In the *cross* sequence, neither SSIM-PF nor DSSIM are distracted by the temporary occlusion of the tracked person by other passer-by, whereas the MS algorithm locks onto a similar object moving in the opposite direction. Likewise, although all the three algorithms manage to follow the target in *man* sequences, the gradient structural similarity method identifies the scale and the position of the object with the best accuracy.

4.4.1 Performance evaluation of the extension of the SSIM-based tracking algorithm

Below, we present a performance analysis of the hybrid structural similarity-based PF algorithm. For the sake of completeness, six competing algorithms has been tested and compared: colour-based PF algorithm COL-PF, SSIM-PF, their hybridised versions, hybrid SSIM-PF-DSSIM (Section 3.5) and hybrid COL-PF-MS (based on procedure proposed in (Lu et al. (2009))), and two deterministic procedures themselves (DSSIM and MS). A discussion of the results based on the visual observation of tracking output in the *cross* sequence is provided below and the specific features of the algorithms tested are pointed out. Figure 6 presents the extracted frames of the output of the six tracking algorithms.
It should be noted that in order to illustrate the benefit of using the optimisation procedures, a very low number of the particles for PF-based methods has been chosen (20 for SSIM-based and 30 for colour-based PF). Consequently, it allowed us to observe whether the resulting tracking instability and failures are partially mitigated by the use of the optimisation procedures. Moreover, since the optimisation procedures are much faster than PFs, such a combination does not increase the computational load considerably. On the contrary, the appropriate combination of the two methods, results in a lower number of the particles required and thus reducing the processing time. Conversely, it can be shown that, a non-optimised tracking algorithms can achieve a similar performance to the optimised tracking algorithm utilising a larger number of particles and thus being more computationally demanding.

Fig. 5. Example video frames with output of the tracking algorithm output superimposed: DSSIM solid blue rectangle, SSIM-PF dashed green rectangle and MS solid magenta rectangle

Based on the observation of the estimated target regions in Figure 6, it can be concluded that the gradient structural similarity procedure locates the object precisely in majority of the frames. It fails, however, to recover from an occlusion towards the end of the sequence. The performance of the SSIM-PF is very unstable, due to a very low number of particles used and it looses the object half-way through the sequence. On the other hand, the combined algorithm, SSIM-PF-DSSIM, tracks the object successfully throughout the sequence. The MS algorithm has completely failed to track the object. Since the MS algorithm is a memory-less colour-based tracking algorithm, its poor performance in these sequence is due to the object's fast motion and its similarity to the surrounding background. The colour-based algorithm, COL-PF, performs similarly to SSIM-PF, however, it locates the object somewhat more precisely. Finally, the combined COL-PF-MS algorithm, appears to be more stable than its non-optimised version. Nevertheless, the objects is eventually lost as a result of the occlusion.

Finally, to illustrate a potential of further extension of the SSIM-PF, a type of target distortion, for which the state space can be easily extended, is considered: rotation of the target in the plane approximately perpendicular to the camera's line-of-sight. A simple solution to the tracking of the rotating objects is to include an orientation of the target in the state space, by taking $\boldsymbol{x} = (x_k, y_k, s_k, \alpha_k)^T$ as the state variable in the algorithm described in Table 2, where α_k is the orientation angle. The complexity of the algorithm is increased slightly due to the need to generate the rotated versions of the reference object (which can, possibly, be pre-computed). For some video sequences it may also be necessary to increase the number of particles, in order to sufficiently sample the state space. The results of tracking a rotating trolley in a sequence from PETS 2006 Benchmark Data (Nin (2006)), with the use of 150 particles are shown in Figure 7. The figure shows examples of frames from two best-performing tracking techniques, 'colour&edges' and 'structure'. Apart from the rotation scaling of the object, additional difficulty in tracking arose because the object was partially transparent and thus often took on the appearance of the non-stationary background. However, also in this case 'structure' tracking algorithm appears to follow the location, scale and rotation of the object more closely than the other algorithms.

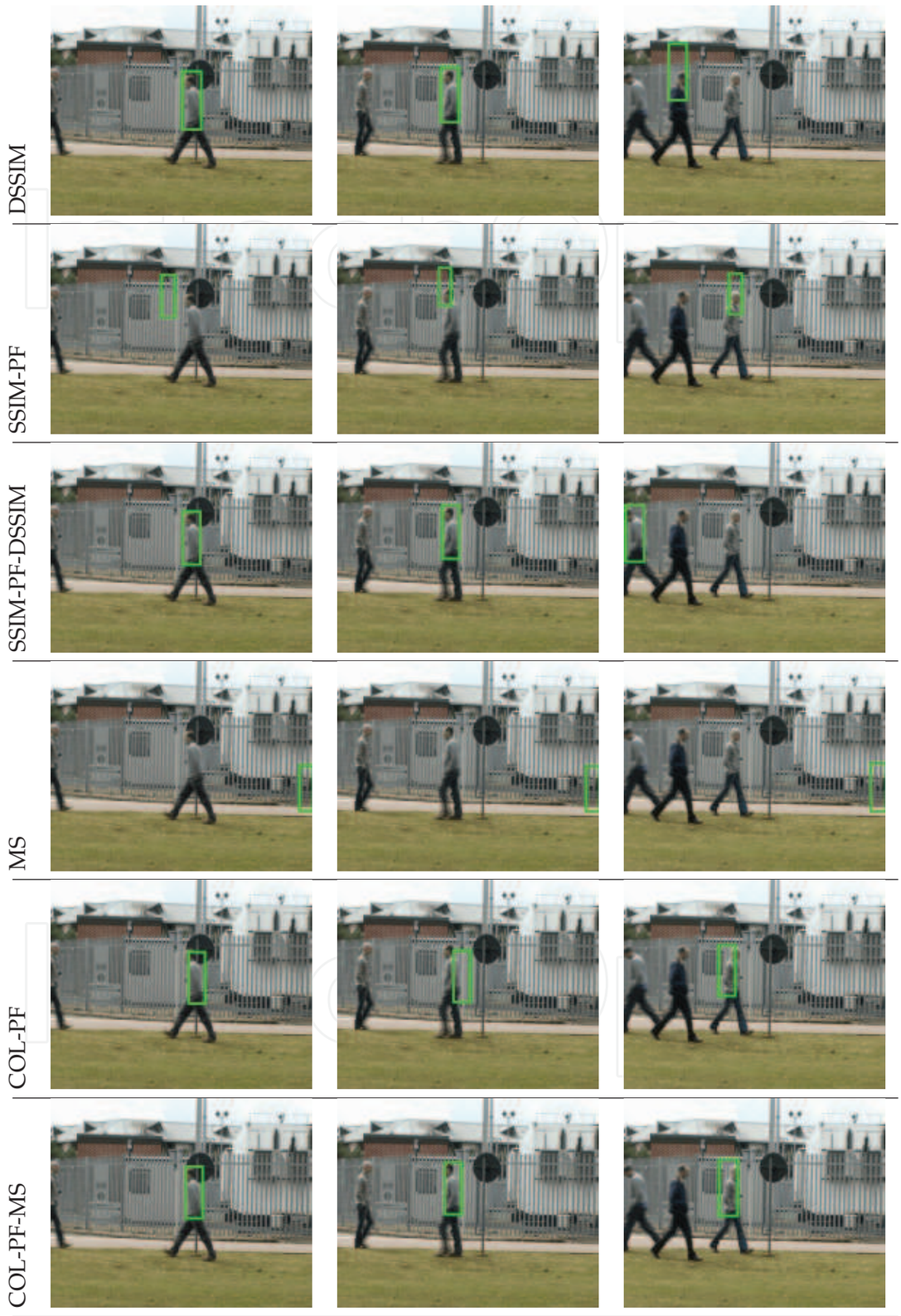


Fig. 6. Pedestrian tracking test results

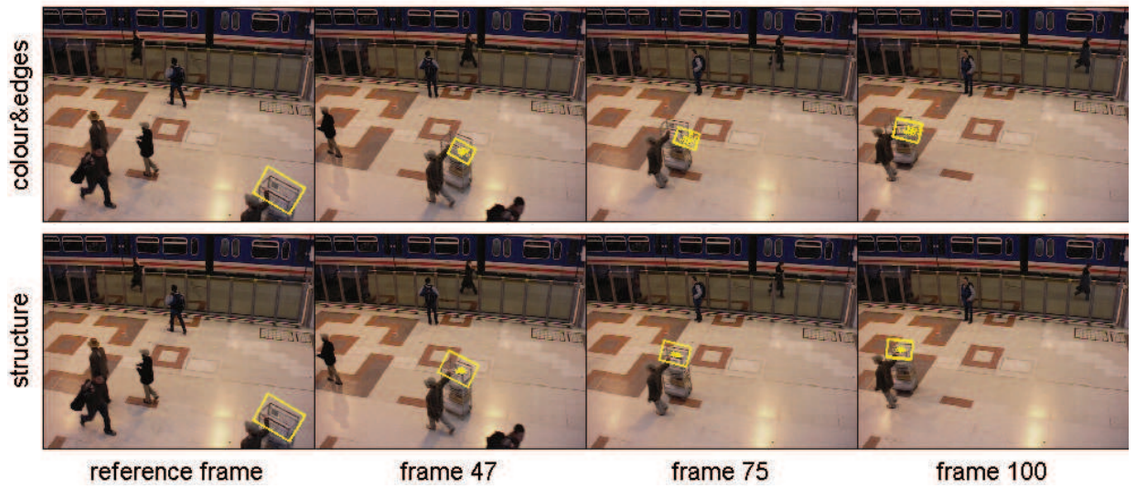


Fig. 7. Example video frames with a tracking output (tracking rectangle and particles) superimposed, sequence *S1-T1-C* containing a rotating object

5. Discussion and conclusions

The recently developed video tracking methods based on structural similarity and their new extensions have been presented in this work. Novel deterministic and hybrid probabilistic approaches to video target tracking have been investigated, and their advantages and mutual complementarities have been identified. First, a fast deterministic procedure that uses the gradient of the structural similarity surface to localise the target in a video frame has been derived. Next, a hybrid PF-based scheme, where each particle is optimised with the use of the aforementioned gradient procedure has been proposed.

The performance of the structural similarity-based methods has been contrasted with selected tracking methods based on colour and edge cues. The structural similarly methods, while being computationally less expensive, perform better, on average, than the colour, edge and mean shift, as shown in the testing surveillance video sequences. Specifically, the results obtained with the hybrid technique proposed indicate that a considerable improvement in tracking is achieved by applying the optimisation scheme, while the price of a moderate computational complexity increase of the algorithm is off-set by the low number of particles required.

The particular issue addressed herein is concerned with tracking object in the presence of spurious or similarly-coloured targets, which may interact or become temporarily occluded. All structural similarity-based method have been shown to perform reliably under difficult conditions (as often occurs in surveillance videos), when tested with real-world video sequences. Robust performance has been demonstrated in both low and variable light conditions, and in the presence of spurious or camouflaged objects. In addition, the algorithm copes well with the artefacts that may be introduced by a human operator, such as rapid changes in camera view angle and zoom. This is achieved with relatively low computational complexity, which makes these algorithms potentially applicable to real-time surveillance problems.

Among the research issues that will be the subject of further investigation is a further speed and reliability improvement of the proposed optimised hybrid technique. It is envisaged that this could be achieved by replacing the simple gradient search with a more efficient

optimisation procedure and by more accurate modelling of the resulting proposal density. The structural similarity measure-based tracker, although giving very precise performance, may in some cases be sensitive to alteration of the tracked object, for example its significant rotation or long occlusion. Thus, the recovery and/or template update techniques will also be investigated in the future to improve reliability of the proposed tracker.

6. Acknowledgements

We would like to thank the support from the [European Community's] Seventh Framework Programme [FP7/2007-2013] under grant agreement No 238710 (Monte Carlo based Innovative Management and Processing for an Unrivalled Leap in Sensor Exploitation) and the EU COST action TU0702. The authors are grateful for the support offered to the project by National Natural Science Foundation of China Research Fund for International Young Scientists and EU The Science & Technology Fellowship Programme in China.

7. References

Aghajan, H. & Cavallaro, A. (2009). *Multi-Camera Networks: Principles and Applications*, Academic Press.

Aherne, F., Thacker, N. & Rockett, P. (1990). The quality of training-sample estimates of the Bhattacharyya coefficient, *IEEE Trans. on PAMI* 12(1): 92–97.

Arulampalam, M., Maskell, S., Gordon, N. & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. on Signal Proc.* 50(2): 174–188.

Bai, K. & Liu, W. (2007). Improved object tracking with particle filter and mean shift, *Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China*, Vol. 2, pp. 221–224.

Bradski, G. (1998). Computer vision face tracking as a component of a perceptual user interface, *Workshop on Applic. of Comp. Vision*, Princeton, NJ, pp. 214–219.

Brasnett, P., Mihaylova, L., Canagarajah, N. & Bull, D. (2005). Particle filtering with multiple cues for object tracking in video sequences, *Proc. of SPIE's 17th Annual Symposium on Electronic Imaging, Science and Technology, V. 5685*, pp. 430–441.

Brasnett, P., Mihaylova, L., Canagarajah, N. & Bull, D. (2007). Sequential Monte Carlo tracking by fusing multiple cues in video sequences, *Image and Vision Computing* 25(8): 1217–1227.

Cai, Y., de Freitas, N. & Little, J. J. (2006). Robust visual tracking for multiple targets, *In Proc. of European Conference on Computer Vision, ECCV*, pp. 107–118.

Chang, C. & Ansari, R. (2003). Kernel particle filter: Iterative sampling for efficient visual tracking, *Proc. of ICIP* 2003, pp. III - 977-80, vol. 2.

Chang, C. & Ansari, R. (2005). Kernel particle filter for visual tracking, *IEEE Signal Processing Letters* 12(3): 242–245.

Chen, D. & Yang, J. (2007). Robust object tracking via online dynamic spatial bias appearance models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29: 2157–2169.

Cheng, Y. (1995). Mean shift, mode seeking and clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17(8): 790–799.

Comaniciu, D. & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(5): 603–619.

Comaniciu, D., Ramesh, V. & Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift, *Proc. of 1st Conf. Comp. Vision Pattern Recogn.*, Hilton Head, SC, pp. 142–149.

Comaniciu, D., Ramesh, V. & Meer, P. (2003). Kernel-based object tracking, *IEEE Trans. Pattern Analysis Machine Intelligence* 25(5): 564–575.

Cvejic, N., Nikolov, S. G., Knowles, H., Loza, A., Achim, A., Bull, D. R. & Canagarajah, C. N. (2007). The effect of pixel-level fusion on object tracking in multi-sensor surveillance video, *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Minneapolis, Minnesota, USA*.

Doucet, A., Freitas, N. & N. Gordon, E. (2001). *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag.

Forsyth, D., Arikan, O., Ikemoto, L. & Ramanan, D. (2006). *Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis. Foundations and Trends in Computer Graphics and Vision*, Hanover, Massachusetts. Now Publishers Inc.

Fukunaga, K. & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition, *Information Theory, IEEE Transactions on* 21(1): 32 – 40.

Gandhi, T. & Trivedi, M. (2007). Pedestrian protection systems: Issues, survey and chllenges, *IEEE Transactions on Intelligent Transportation Systems* 8(3): 413–430.

Gerónimo, D., López, A. M., Sappa, A. D. & Graf, T. (2010). Survey of pedestrian detection for advanced driver assistance systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32: 1239–1258.

Han, B., Comaniciu, D., Zhu, Y. & Davis, L. S. (2004). Incremental density approximation and kernel-based bayesian filtering for object tracking., *CVPR (1)*, pp. 638–644.

Hu, W., Tan, T., Wang, L. & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors, *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 34(3): 334 –352.

Isard, M. & Blake, A. (1996). Contour tracking by stochastic propagation of conditional density, *European Conf. on Comp. Vision*, Cambridge, UK, pp. 343–356.

Isard, M. & Blake, A. (1998). Condensation – conditional density propagation for visual tracking, *Intl. Journal of Computer Vision* 28(1): 5–28.

Khan, Z., Balch, T. & Dellaert, F. (2005). MCMC-based particle filtering for tracking a variable number of interacting targets, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(11): 1805–1819.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *J. Comput. Graph. Statist.* 5(1): 1Ű25.

Koch, W. (2010). On Bayesian tracking and data fusion: A tutorial introduction with examples, *IEEE Transactions on Aerospace and Electronics Magazine Part II: Tutorials* 7(July): 29–52.

Lewis, J. J., O'Callaghan, R. J., Nikolov, S. G., Bull, D. R. & Canagarajah, C. (2007). Pixel- and region-based image fusion using complex wavelets, 8(2): 119–130.

Liu, J. & Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association* 93(443): 1032–1044.

Loza, A., Mihaylova, L., Bull, D. R. & Canagarajah, C. N. (2006). Structural similarity-based object tracking in video sequences, *Proc. Intl. Conf. on Information Fusion, Florence, Italy*, pp. 1–6.

Loza, A., Mihaylova, L., Bull, D. R. & Canagarajah, C. N. (2009). Structural similarity-based object tracking in multimodality surveillance videos, *Machine Vision and Applications* 20(2): 71–83.

Lu, W.-L., Okuma, K. & Little, J. J. (2009). Tracking and recognizing actions of multiple hockey players using the boosted particle filter, *Image Vision Comput.* 27(1-2): 189–205.

Maggio, E. & Cavallaro, A. (2005). Hybrid particle filter and mean shift tracker with adaptive transition model, *Proc. of ICASSP* 2005, pp. 221-224, vol. 2.

Mihaylova, L., Loza, A., Nikolov, S. G., Lewis, J., Canga, E. F., Li, J., Bull, D. R. & Canagarajah, C. N. (2006). The influence of multi-sensor video fusion on object tracking using a particle filter, *Proc. Workshop on Multiple Sensor Data Fusion, Dresden, Germany*, pp. 354–358.

Nin (2006). PETS 2006 benchmark data, Dataset available on-line at: http://www.pets2006.net.

Nummiaro, K., Koller-Meier, E. B. & Gool, L. V. (2003). An adaptive color-based particle filter, *Image and Vision Computing* 21(1): 99–110.

Okuma, K., Taleghani, A., de Freitas, N., Little, J. & Lowe, D. (2004). A boosted particle filter: Multitarget detection and tracking, *In Proc. of European Conference on Computer Vision*, Vol. 1, pp. 28–39.

PerceptiVU, Inc. (n.d.). Target Tracking Movie Demos. http://www.perceptivu.com/MovieDemos.html.

Pérez, P., Vermaak, J. & Blake, A. (2004). Data fusion for tracking with particles, *Proceedings of the IEEE* 92(3): 495–513.

Ristic, B., Arulampalam, S. & Gordon, N. (2004). *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Boston, London.

Shan, C., Tan, T. & Wei, Y. (2007). Real-time hand tracking using a mean shift embedded particle filter, *Pattern Recogn.* 40(7): 1958–1970.

Shen, C., van den Hengel, A. & Dick, A. (2003). Probabilistic multiple cue integration for particle filter based tracking, *Proc. of the VIIth Digital Image Computing : Techniques and Applications*, C. Sun, H. Talbot, S. Ourselin, T. Adriansen, Eds.

Smith, D. & Singh, S. (2006). Approaches to multisensor data fusion in target tracking: A survey, *IEEE Transactions on Knowledge and Data Engineering* 18(12): 1696–1710.

The Eden Project Multi-Sensor Data Set (2006). http://www.imagefusion.org/.

Triesch, J. & von der Malsburg, C. (2001). Democratic integration: Self-organized integration of adaptive cues, *Neural Computation* 13(9): 2049–2074.

Wan, E. & van der Merwe, R. (2001). *The Unscented Kalman Filter, Ch. 7: Kalman Filtering and Neural Networks. Edited by S. Haykin*, Wiley Publishing, pp. 221–280.

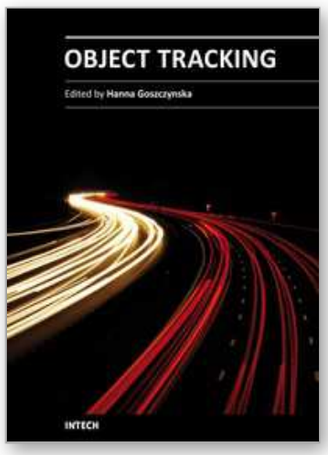
Wang, Z., Bovik, A. C. & Simoncelli, E. P. (2005a). Structural approaches to image quality assessment, *in* A. Bovik (ed.), *Handbook of Image and Video Processing, 2nd Edition*, Academic Press, chapter 8.3.

Wang, Z., Bovik, A., Sheikh, H. & Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing* 13(4): 600–612.

Wang, Z. & Simoncelli, E. (2004). Stimulus synthesis for efficient evaluation and refinement of perceptual image quality metrics, *IS & T/ SPIE's 16th Annual Symposium on Electronic Imaging. Human Vision and Electronic Imaging IX, Proc. SPIE, Vol. 5292*, San Jose, pp. 18–22.

Webb, A. (2003). *Statistical Pattern Recognition*, John Wiley & Sons.

Zhao, Q., Brennan, S. & Tao, H. (2007). Differential EMD tracking, *Proc. of IEEE 11th International Conference on Computer Vision*, pp. 1–8.



Object Tracking

Edited by Dr. Hanna Goszczynska

ISBN 978-953-307-360-6
Hard cover, 284 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

Object tracking consists in estimation of trajectory of moving objects in the sequence of images. Automation of the computer object tracking is a difficult task. Dynamics of multiple parameters changes representing features and motion of the objects, and temporary partial or full occlusion of the tracked objects have to be considered. This monograph presents the development of object tracking algorithms, methods and systems. Both, state of the art of object tracking methods and also the new trends in research are described in this book. Fourteen chapters are split into two sections. Section 1 presents new theoretical ideas whereas Section 2 presents real-life applications. Despite the variety of topics contained in this monograph it constitutes a consisted knowledge in the field of computer object tracking. The intention of editor was to follow up the very quick progress in the developing of methods as well as extension of the application.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Artur Loza, Lyudmila Mihaylova, Fanglin Wang and Jie Yang (2011). Structural Information Approaches to Object Tracking in Video Sequences, Object Tracking, Dr. Hanna Goszczynska (Ed.), ISBN: 978-953-307-360-6, InTech, Available from: http://www.intechopen.com/books/object-tracking/structural-information-approaches-to-object-tracking-in-video-sequences

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen