

Research Article

Video Tracking Using Dual-Tree Wavelet Polar Matching and Rao-Blackwellised Particle Filter

**Sze Kim Pang, James D. B. Nelson, Simon J. Godsill (EURASIP Member),
and Nick Kingsbury (EURASIP Member)**

*Signal Processing and Communications Laboratory, Engineering Department, Cambridge University, Trumpington Street,
Cambridge CB2 1PZ, UK*

Correspondence should be addressed to Sze Kim Pang, skp31@cam.ac.uk

Received 12 March 2009; Revised 7 August 2009; Accepted 31 December 2009

Recommended by Abdelak Zoubir

We describe a video tracking application using the dual-tree Polar Matching Algorithm. We develop the dynamical and observation models in a probabilistic setting and study the empirical probability distribution of the Polar Matching output. We model the visible and occluded target statistics using Beta distributions. This is incorporated into a Track-Before-Detect (TBD) solution for the overall observation likelihood of each video frame and provides a principled derivation of the observation likelihood. Due to the nonlinear nature of the problem, we design a Rao-Blackwellised Particle Filter (RBPF) for the sequential inference. Computer simulations demonstrate the ability of the algorithm to track a simulated video moving target in an urban environment with complete and partial occlusions.

Copyright © 2009 Sze Kim Pang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Detection and tracking of a known target in video sequences is a common and important problem in image processing. In this paper, we focus on the scenario of an unmanned air vehicle (UAV) platform-based image sensor as it attempts to track a ground vehicle traversing a cluttered urban environment. The objective is to provide a good estimate of the position and velocity of the vehicle in grid coordinates, and to be robust against temporary occlusions.

As the location of the UAV and target vary, and as the bearing and azimuth of the sensor change, the image of the target will appear to shift and rotate, and possibly change in scale. In this context, it therefore makes sense that any successful detection method must have robustness or invariance to spatial shifts, rotations, and some scale variations.

Wavelets possess several attractive properties for object detection. The fast wavelet transform is order n and represents object features in a sparse, highly localised, and multiresolution form. However the use of wavelets in object detection has been stymied by their lack of shift invariance. When an object is translated, the magnitude of the wavelet

coefficients of the object will change. This has prompted variants of the “spin-cycle” method of Coifman and Donoho [1]; whereby extra-training samples are created by shifting the original ones.

Multiwavelets have been used in object recognition where 1D wavelet-based affine invariant measures are applied to the boundaries of objects [2]. However, this assumes that the boundary of the target is reasonably different from the surrounding clutter. A significant amount of information such as colour and texture is discarded and it also relies upon robust edge detection and edge grouping.

Unlike regular wavelet and multiwavelet constructions, the dual-tree complex wavelet transform (DTCWT) is shift invariant and, at each scale, decomposes an image into six complex, and highly directional subbands [3]. Dual-tree wavelets have a striking similarity to the way that the primary (V1) cortical filters of the human visual system work. They interpolate well in space, orientation, and scale. In fact dual-tree wavelets are a special case of a multiwavelet with very good frequency domain properties. Since the DTCWT has good shift invariance, the highly redundant spin cycle or undecimated transforms are not necessary.

Extending his work on the shift-invariant dual-tree complex wavelet transform, Kingsbury recently introduced the rotation-invariant Polar Matching Algorithm [4]. Owing to low redundancy, the DTCWT descriptor is more efficient than the existing popular scale- and rotation-invariant methods of the Shift Invariant Feature Transform (SIFT) [5] and Simoncelli's steerable pyramids [6]. It is adapted here to provide image matching between a small template and a larger image rather than matching keypoints of two similarly sized images, as previously reported. The output of the Polar Matching Algorithm gives a detection confidence (or likelihood value) of the target of interest for a specific position and orientation within the video frame.

Many approaches have been proposed to tackle the problem of target tracking. These range from Kalman filter and its nonlinear extensions to Joint Probabilistic Data Association Filters (JPDAFs) [7, 8]. The detection output of the Polar Matching Algorithm can be fed into a tracking filter to provide estimates of the target's state. However, an optimal linear filter such as the Kalman filter may not work as well in this scenario. One reason is due to the nonlinear measurement process of the imaging sensor and the Polar Matching Algorithm. Another reason is that the posterior distribution is likely to be multimodal due to the nature of the video data. To overcome these issues, we make use of optimal nonlinear filtering techniques based on sequential Monte Carlo methods [9–11], also known as particle filters, to perform the tracking.

Particle filters have been widely applied in many tracking applications [12–14]. In video tracking, the visual environment is typically fraught with complex features and uncertainties. It is necessary to develop principled probabilistic models to detect and track the target successfully through the cluttered environment. Early work such as BraMBLe by Isard and MacCormick [15] demonstrated the flexibility of the particle filter to perform video tracking of people through a static background. In [16], Vermaak et al. used a mixture particle filter to track hockey players in a video sequence. By distributing the resampling step to individual filters, the mixture particle filter is able to better maintain the posterior multimodality. Khan et al. [17] used an MCMC-based particle filter to track interacting targets. They made use of a Markov Random Field (MRF) prior to help maintain the identity of targets throughout an interaction, significantly reducing tracker failures. In this paper, we will combine the desirable detector characteristics of the Polar Matching Algorithm, with a particle filter in a principled probabilistic fashion. A Rao-Blackwellised Particle Filter (RBPF) will be developed that improves the efficiency of the particles simulation through Rao-Blackwellising the visibility variable. This is then applied to a video sequence to track a vehicle that undergoes occlusions as it moves through a cluttered urban scene.

The paper is organised as follows. Section 2 presents rotation-invariant dual-tree complex wavelet Polar Matching. Section 3 describes the probabilistic state-space model. Sections 4 and 5 describe the dynamic models and observation model, respectively. Section 5 describes the particle filter algorithm, and Section 7 describes the Rao-Blackwellised

Particle Filter. Simulation results are discussed in Section 8, followed by our conclusions in Section 9.

2. Polar Matching

The Polar Matching method proceeds by firstly computing the DTCWT coefficients of a template. The complex wavelet coefficients at each of the six directional subbands at the centre point are stored together with their complex conjugates. Coefficients are also taken around one or more circles, about the centre point, at 30° increments from different directional subbands and at multiple scales. As Figure 1 illustrates, the coefficients are then arranged into a Polar Matching matrix (P -matrix) such that a rotation of $k \times 30^\circ$ in the original image will produce a vertical shift by k rows in the P -matrix. Consider two images, one an $n \times 30^\circ$ rotated version of the other; then a sum of column-wise correlations between the two corresponding P -matrices will result in a response curve, with respect to relative rotation angle, and a maximum at location n .

However, the rotational sensitivity can be increased to 7.5° by careful band-limited interpolation. This is achieved by performing the correlation as a product in the Fourier domain and by zero padding. Care should be taken here over the phase rotations of the complex coefficients. The first column of a P -matrix, formed about the centre of a single step edge, will vary slowly as the edge is rotated. Columns 2 and 7 will vary in phase quicker, 3 and 6 quicker still, and 4 and 5 the quickest of all. Hence, the zero padding must be placed according to the P -matrix column number. Coefficients obtained from other scales, or colour bands, can be added by appending them as extra columns to the P -matrix. Hence, this Polar Matching Algorithm takes the property of shift invariance from the DTCWT and approximates rotation invariance from the P -matrix construction. As explained further in [4], the matching operator is rotation invariant in the sense that we only need to construct a single P -matrix for the template and the correlation score output of all the required angles of orientation can be obtained via a single operation using Polar Matching. Note that the orientation of the target is, a priori, unknown and can change over time. Therefore, even when the sensor position is known, a rotation invariant matcher is still required. By the careful radial sampling and ordering of the directional DTCWT subband coefficients, Polar Matching transforms rotations in the object into shifts in the feature vectors. We can contrast this with a naive template correlation matching, where the template is rotated by each angle before the correlation score is calculated. In Figure 2, we plot the result of correlating the original template with rotated copies. The maximum of each curve indicates the degree of match and its location correctly gives the object orientation.

We define the column vector $\mathbf{h} \in \mathbb{C}^n$ as the concatenation of the Fourier transform of the P -matrix columns. Likewise, let $\mathbf{f} = \mathbf{f}(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{C}^n$ be the concatenation of Fourier coefficients of the P -matrix of a test image taken at the point $\mathbf{x} = (\tilde{x}, \tilde{y})$. Together, the inverse Fourier transform and zero padding can be described as a block diagonal matrix operator

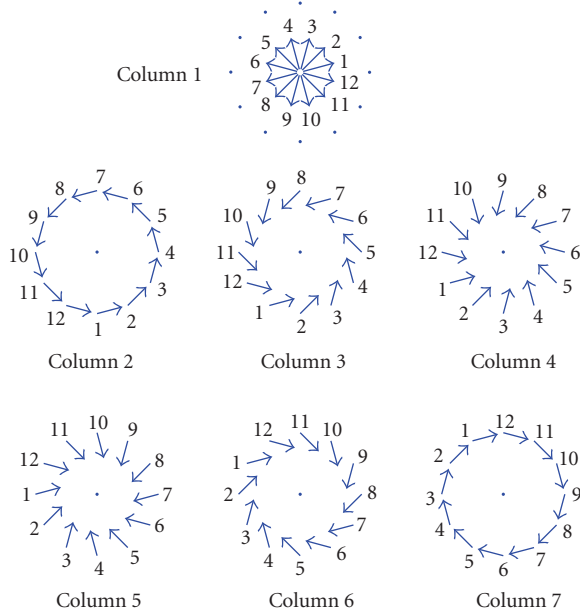


FIGURE 1: Locations and orientations of the DTCWT coefficients. Each orientation describes a coefficient, or conjugate, of one of the six subbands. The column numbers indicate the column of the P -matrix in which the coefficients are placed, and the numbers displayed around each circle indicate the row of the P -matrix that each coefficient is placed. Taken from [4].

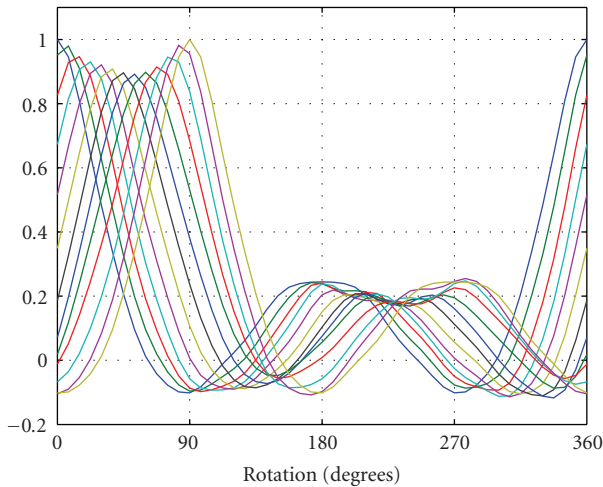


FIGURE 2: Correlation results for the template shown in Figure 3: each curve represents the Polar Matching output obtained by correlating the template with rotated versions of itself, in 5° increments from 0° to 90° . For each curve, the output is a response with respect to 48 angles in 7.5° increments.

W. Then, the Polar Matching operation at the angle θ can be written as

$$(\mathbf{h} \star \mathbf{f})(\theta) = \frac{\Re(\mathbf{h}^H \mathbf{W}(\theta) \mathbf{f})}{\|\mathbf{h}\|_2 \|\mathbf{f}\|_2}, \quad (1)$$

where the superscript H denotes conjugate, or Hermitian, transpose, and the real component \Re is taken to return

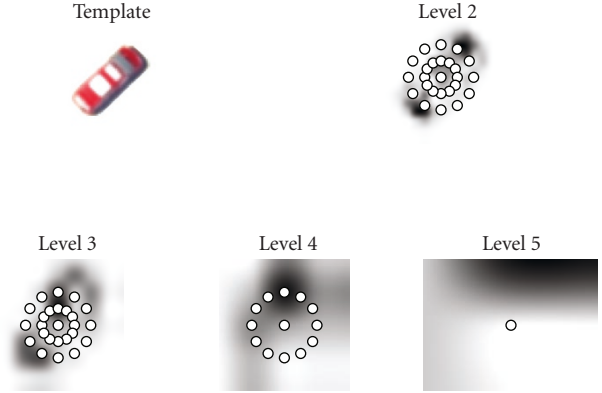


FIGURE 3: Magnitude of the template DTCWT coefficients. The white dots indicate the locations of the coefficients that are used in the P -matrix template. Four different scales, or levels, are used. Only one of the 6 subbands is shown.

the correlation intensity. As Figure 3 shows, 13 complex coefficients are taken from levels 2 and 3, 7 from level 4, and one from level 5. This results in $n = 34 \times 12\zeta$ coefficients, where ζ is the zero padding factor in the Fourier domain ($\zeta = 4$ was used in our experiments). Note that only 34×12 of these coefficients are nonzero.

In practice, the feature vectors are normalised to have unit ℓ_2 -norm, the numerator of (1) is merely a weighted dot product and we have that

$$(\mathbf{h} \star \mathbf{f})(\theta) = \Re \left(\sum_{i=1}^n w_i(\theta) \bar{h}_i f_i \right), \quad (2)$$

where h_i is the i th element of \mathbf{h} , and the bar denotes complex conjugation.

A simple method for finding the target's location and orientation can be done by computing $\arg \max_{\mathbf{x}, \theta} (\mathbf{h} \star \mathbf{f}(\mathbf{x}))(\theta)$. Colour images, and coefficients from other scales, are dealt with by simply concatenating the P -matrix Fourier coefficients of each colour channel into the vectors \mathbf{h} and \mathbf{f} . In this work, we use the RGB colour bands for each of the scale levels and sampling strategies shown in Figure 3.

In summary, Polar Matching provides a correlation score between -1 and 1 for a template in a specific position (\tilde{x}, \tilde{y}) and orientation θ within a larger video frame. In the following, the Polar Matching function is referred to as $\text{Polar}(\tilde{x}, \tilde{y}, \theta)$. For the video tracking application, the template will be provided as a P -matrix, taken directly from an image of the target vehicle.

3. Bayesian Filtering

We now develop a probabilistic framework for the single target video tracking problem, using the Polar Matching Algorithm in the previous section as a detector.

We are interested in the target's position (x_t, y_t) and velocity (\dot{x}_t, \dot{y}_t) in grid coordinates in the real world at time t . The target's position in the image plane $(\tilde{x}_t, \tilde{y}_t)$ of the video sensor can be obtained by a nonlinear transformation $H(\cdot)$.

This maps the target's real world coordinates to the image coordinates using the sensor's position and orientation. This is given by

$$(\tilde{x}_t, \tilde{y}_t) = H(x_t, y_t). \quad (3)$$

In this paper, we will assume that the sensor's position and orientation are known; that is, $H(\cdot, \cdot)$ is known. Also, $\tilde{x}_t \in [\tilde{x}_{\min}, \tilde{x}_{\max}]$ and $\tilde{y}_t \in [\tilde{y}_{\min}, \tilde{y}_{\max}]$. In the video example considered in this paper, $\tilde{x}_{\min} = 1$ and $\tilde{x}_{\max} = 1280$, which is the number of horizontal pixels. Similarly, $\tilde{y}_{\min} = 1$ and $\tilde{y}_{\max} = 720$ which is the number of vertical pixels.

We also will like to know the orientation of the image template, $\theta_t \in (\theta_{\min}, \theta_{\max})$, with respect to each video frame at time t . This is important in computing the observation output based on the Polar Matching Algorithm. Again, in this paper's video example, we use $\theta_{\min} = -180^\circ$ and $\theta_{\max} = 180^\circ$.

Furthermore, the target of interest may be fully or partially occluded due to buildings or other visual occlusions such as smoke. We will introduce a visibility variable $V_t \in \{0, 1\}$ to model this; that is, when the target is visible, $V_t = 1$. This approach in modelling visibility is similar to the approach in modelling existence variable found in Track-Before-Detect (TBD) problems [12–14]. In the TBD approach, the existence variable is typically assumed to be independent of the position of the target, and this can model targets disappearing from the surveillance scene at anytime. Similarly, the visibility variable V_t in this paper is assumed to be independent of the position of the target. This can model targets that can be occluded from the scene at anytime.

Therefore, the joint state at time t is given by $S_t = [x_t \ \dot{x}_t \ y_t \ \dot{y}_t \ \theta_t \ V_t]$. For convenience later, we also define the position and velocity of the target as $X_t = [x_t \ \dot{x}_t \ y_t \ \dot{y}_t]$.

The observation consists of a sequence of video frames captured at time interval T . We will also use T as the rate of change in the dynamic state. However, more generally, state evolution need not be constrained by the observation interval, but evolves with its own dynamics. This can be done in a more flexible model based on variable rate formulation [18, 19].

For each video frame, we will use the Polar Matching Algorithm as a detector. We will assume that we have a template of the target of interest. In order to derive the observation model later, conceptually we will discretise the possible range of image position $(\tilde{x}_t, \tilde{y}_t)$ and orientation θ_t . For each of the discretised triplets represented by $\Delta r = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\theta)$, we can compute the detector output of the Polar Matching Algorithm using $Z_{t,\Delta r} = \text{Polar}(\Delta\tilde{x}, \Delta\tilde{y}, \Delta\theta)$ at time t . For simplicity, we treat negative correlation numbers from the Polar Matching output as zero correlation, that is, $Z_{t,\Delta r} \in [0, 1]$. Hence, the observation Z_t at time t is the collection of all the Polar Matching Algorithm output at all of the discretised positions

$$Z_t = \{Z_{t,\Delta r} \ \forall \Delta r\}. \quad (4)$$

Assuming a Markovian state transition, the standard state update and prediction equations are given by

$$p(S_t | Z_{1:t}) = \frac{p(Z_t | S_t)p(S_t | Z_{1:t-1})}{p(Z_t | Z_{1:t-1})}, \quad (5)$$

$$p(S_t | Z_{1:t-1}) = \int p(S_t | S_{t-1})p(S_{t-1} | Z_{1:t-1})dS_{t-1},$$

with $Z_{1:t} = [Z_1 \ \dots \ Z_m \ \dots \ Z_t]$. Note that the observation Z_t is typically modelled as conditionally independent of $Z_{1:t-1}$ given the state S_t in similar type of tracking problems [12–14].

4. Dynamical Models

We choose to write the transition probability model as

$$p(S_t | S_{t-1}) = p(X_t | X_{t-1})p(\theta_t | \theta_{t-1})p(V_t | V_{t-1}). \quad (6)$$

The state variables X_t , θ_t , and V_t are modelled to be independent of each other. It is also possible to make the orientation θ_t partially dependent on the target's position and velocity. Here, for simplicity, we use the independent model.

For the target dynamic, we use the discrete time equivalent of the near constant velocity model [20]. This can be used to track target with small amount of acceleration—with the acceleration being essentially modeled by the random noise term in the model. It can model gradual changes to velocity, which is the case for the example considered here in this paper. For targets that maneuver more aggressively, other more complex models such as semi-Markov models are available [19, 20].

The dynamical model for X_t is given by

$$X_t = FX_{t-1} + w_t, \quad (7)$$

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where w_t is a Gaussian noise with covariance Q_w given by

$$Q_w = \begin{bmatrix} \frac{T^3}{3}q_x & \frac{T^2}{2}q_x & 0 & 0 \\ \frac{T^2}{2}q_x & Tq_x & 0 & 0 \\ 0 & 0 & \frac{T^3}{3}q_y & \frac{T^2}{2}q_y \\ 0 & 0 & \frac{T^2}{2}q_y & Tq_y \end{bmatrix} \quad (8)$$

with q_x and q_y being the variance of the driving noise of the dynamical process. Hence,

$$X_t | X_{t-1} \sim N(FX_{t-1}, Q_w). \quad (9)$$

The orientation θ_t allows us to track the vehicle as it rotates in plane within the image. This is modelled as a random walk:

$$\theta_t | \theta_{t-1} \sim N(\theta_{t-1}, q\theta), \quad (10)$$

where $q\theta$ is the variance of the driving noise.

The visibility variable V_t determines whether the target is visible or if it is temporarily obscured by smoke or walls. This affects the way the likelihood term $p(Z_t | S_t)$ behaves. The target's visibility variable will be modelled as a discrete Markov chain, similar to the existence variable in the TBD approach [12–14]. For example, at each time t , there is a probability P_{NV} that a target previously being visible at time $t - 1$ can become occluded at time t . This is summarised by the following equations:

$$\begin{aligned} p(V_t = 0 | V_{t-1} = 1) &= P_{NV}, \\ p(V_t = 1 | V_{t-1} = 0) &= P_V. \end{aligned} \quad (11)$$

5. Observation Model Using Polar Matching

In this section, we will construct a more principled derivation of the observation model $p(Z_t | X_t, \theta_t, V_t)$. We note that each video frame detection Z_t obtained using the Polar Matching Algorithm consists of measurements of a 3-dimensional discretised grid with coordinates represented by $\Delta r = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\theta)$. Here, $\Delta\tilde{x}$ belongs to the set of discretised range $\{\tilde{x}_{\min}, \tilde{x}_{\min} + \delta\tilde{x}, \dots, \tilde{x}_{\max}\}$ for some $\delta\tilde{x}$. The ranges of $\Delta\tilde{y}$ and $\Delta\theta$ are similarly defined.

It is important to note that we are not here dealing with point spatial measurements, as would be the case for standard radar and sonar detectors; instead we receive at each time frame t a fully populated 3D pixellated grid of measurements, with each pixel giving a detection intensity corresponding to a particular pixel's spatial location and orientation. We will assume that the target of interest will always be within the video frame, but it may or may not be visible, and it can occupy only one of the positions in the 3D grid. Again, this approach is similar to the TBD type of observation models that are found in the literature [12–14]. This is an approximation as it is possible for the target to occupy more than one grid point, depending on how the grid is constructed or discretisation is done, and possible future extensions of our model would include a point spread function to model target extents of greater than one pixel.

The grid measurements are modelled as conditionally independent given the states X_t , θ_t , and V_t . As stated above, the polar matching detector gives us a discrete (pixellated) grid of measurements Z_t . We are also assuming that a visible target affects only the single 3D pixel cell in which it is located. Hence, for a given target location and orientation θ_t and X_t , we first compute which pixel the target lies in, using (3) to map between spatial position/orientation and image plane pixel location—denote this pixel Δr . The corresponding measurement at this pixel location, $Z_{t,\Delta r}$, is then assumed to be drawn according to a specially learned visible target (foreground) Probability Density Function (PDF) $p(Z_{t,\Delta r} | V_t = 1)$. All remaining pixels in measurement grid Z_t are then assumed to be drawn from a “background” PDF

$p(Z_{t,\Delta r'} | V_t = 0)$, again specially learned from training data. As we will see later in this section, both background and visible target PDFs are modelled by Beta distributions with appropriately fitted parameters. Now, since the observations are considered conditionally independent given the target parameters, we arrive at a product of (one single) foreground PDF at pixel location Δr and many background PDFs at the remaining pixels:

$$\begin{aligned} p(Z_t | X_t, \theta_t, V_t = 1) &= p(Z_{t,\Delta r} | V_t = 1) \times \prod_{\forall \Delta r', \Delta r' \neq \Delta r} p(Z_{t,\Delta r'} | V_t = 0) \\ &= \Omega \frac{p(Z_{t,\Delta r} | V_t = 1)}{p(Z_{t,\Delta r} | V_t = 0)}, \end{aligned} \quad (12)$$

where $p(Z_{t,\Delta r} | V_t = 1)$ is the visible target PDF and $p(Z_{t,\Delta r} | V_t = 0)$ is the occluded target/background PDF. Here it is assumed that occluded target locations have the same distribution as background locations. The proportionality constant Ω is given by

$$\Omega = \prod_{\forall \Delta r'} p(Z_{t,\Delta r'} | V_t = 0). \quad (13)$$

Similarly, if the target is at the Δr grid position, but occluded from view of the sensor, that is, $V_t = 0$, we assume that all pixels are drawn from the background PDF, that is,

$$\begin{aligned} p(Z_t | X_t, \theta_t, V_t = 0) &= p(Z_{t,\Delta r} | V_t = 0) \times \prod_{\forall \Delta r', \Delta r' \neq \Delta r} p(Z_{t,\Delta r'} | V_t = 0) \\ &= \Omega. \end{aligned} \quad (14)$$

Since the proportionality constant Ω is the same for the above two observation likelihoods, we can write

$$p(Z_t | X_t, \theta_t, V_t = 1) \propto \frac{p(Z_{t,\Delta r} | V_t = 1)}{p(Z_{t,\Delta r} | V_t = 0)}, \quad (15)$$

$$p(Z_t | X_t, \theta_t, V_t = 0) \propto 1. \quad (16)$$

The above provides a simple framework for the detection of a single target using a template-based detector. An advantage of this approach is that it is not necessary to compute the Polar Matching output of the entire video frame in order to compute the observation likelihood. Furthermore, the actual discretisation of the grid is not important, as long as we can compute $p(Z_{t,\Delta r} | V_t)$. We can make the discretisation as fine as we want. In practice, we can just directly use X_t and θ_t without any discretisation to compute $p(Z_{t,\Delta r} | V_t)$.

Now, we need to determine the visible target PDF $p(Z_{t,\Delta r} | V_t = 1)$ and the occluded target/background PDF $p(Z_{t,\Delta r} | V_t = 0)$, which are the probabilities of observing the Polar Matching output $Z_{t,\Delta r} = \text{Polar}(\Delta\tilde{x}, \Delta\tilde{y}, \Delta\theta)$ with a specific template. As we do not have an analytical probabilistic

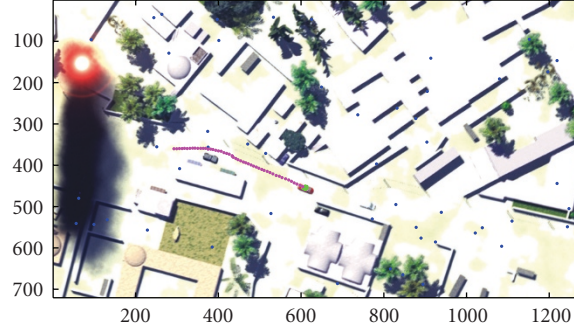


FIGURE 4: Green and blue dots show the sampling for the visible target and occluded target/background histogram, respectively. Magenta dots show the true target path projected onto current video frame.

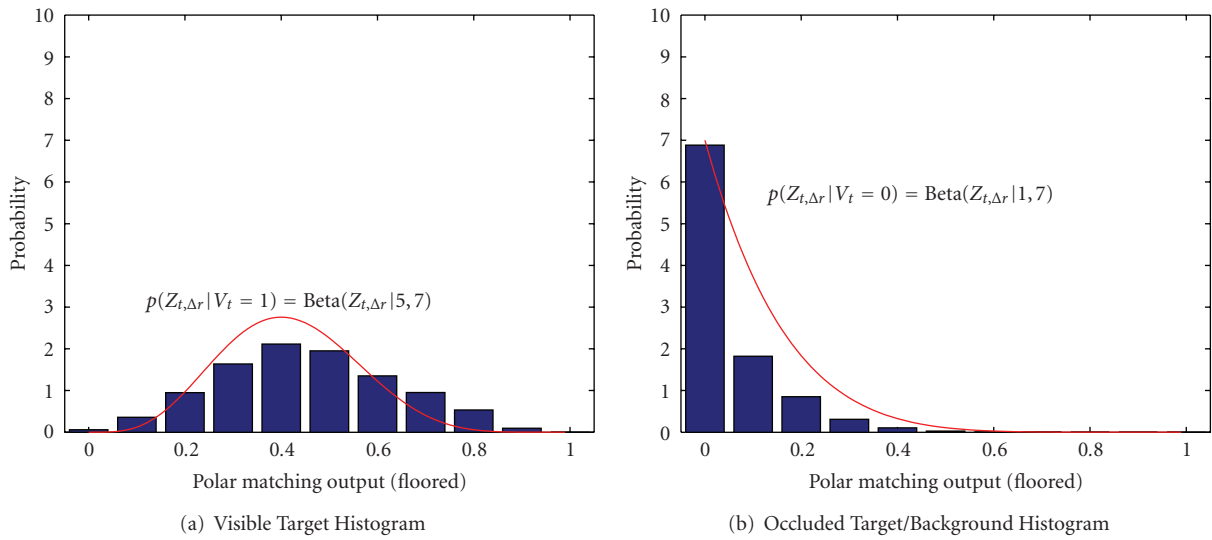


FIGURE 5: These figures show the Visible and Occluded Target/Background histograms for 100 video frames and their respective Beta distributions chosen to model them.

model of the output of the Polar Matching Algorithm, we will use labeled samples of occluded/unoccluded data to analyse the type of empirical densities obtained from the algorithm. Conceptually, these densities are similar to the observation densities used in traditional tracking system such as radar. For example, $p(Z_{t,\Delta r} | V_t = 0)$ is similar in concept to the Rayleigh distributed clutter density when a target is not present in the radar echo [7].

We will now study the histogram of the Polar Matching output using 100 video frames in order to determine the two densities. For the occluded target/background PDF, we will sample random positions in each of the video frame and pass it to the Polar Matching Algorithm to obtain $Z_{t,\Delta r}$. We can see this sampling done in one of the particular video frames in Figure 4. Background sampled locations for a single frame are shown as blue dots. Figure 5(b) shows the histogram for the occluded target/background PDF. Similarly, for the visible target PDF, they are randomly sampled within 5 pixels of the actual target position. Due to the lack of ground truth data, actual target locations are identified by hand-annotation of the video sequence. The sampled points for

the visible target can be seen as green dots in Figure 4. Figure 5(a) shows the histogram for the visible target PDF.

We now attempt to find a parametric model for the distribution of visible target and occluded target/background points. As the range of $Z_{t,\Delta r}$ lies between 0 and 1, a simple distribution such as the Beta distribution can be used to fit the histograms. It is a family of continuous probability distributions defined on the interval $[0, 1]$ parameterized by two positive shape parameters, typically denoted by α and β . By changing α and β , a wide range of shape of distributions can be obtained.

The Beta probability density function is given by

$$\begin{aligned} \text{Beta}(s | \alpha, \beta) &= \frac{(s)^{\alpha-1}(1-s)^{\beta-1}}{\int_0^1 (s)^{\alpha-1}(1-s)^{\beta-1} ds} \\ &= \frac{1}{B(\alpha, \beta)} (s)^{\alpha-1}(1-s)^{\beta-1}, \end{aligned} \quad (17)$$

where $B(\alpha, \beta)$ is the Beta function.

We use the Beta distribution to model both the visible target and occluded target/background histograms. The parameters α and β are selected so that the shape of the distributions fit the histograms, as well as the modes of the distributions are close to the actual modes of the histograms. This is shown in Figure 5. Hence,

$$\begin{aligned} p(Z_{t,\Delta r} | V_t = 1) &= \text{Beta}(Z_{t,\Delta r} | 5, 7), \\ p(Z_{t,\Delta r} | V_t = 0) &= \text{Beta}(Z_{t,\Delta r} | 1, 7). \end{aligned} \quad (18)$$

Substituting the above Beta distributions into (15) gives

$$\begin{aligned} p(Z_t | X_t, \theta_t, V_t = 1) &\propto \frac{\text{Beta}(Z_{t,\Delta r} | 5, 7)}{\text{Beta}(Z_{t,\Delta r} | 1, 7)} \\ &= \frac{B(1, 7)}{B(5, 7)} (Z_{t,\Delta r})^4. \end{aligned} \quad (19)$$

6. Particle Filter Algorithm

The filtering distribution of the dynamical and observation probability model above is complex and nonlinear. Sequential Monte Carlo methods such as particle filters [10–12] can be used to do the inference. A particle filter represents the required posterior density function by a set of random samples (or particles) with associated weights $\{S_{t,p}, w_{t,p}\}_{p=1}^{N_p}$. These particles are then propagated through time to give predictions of the posterior distribution function at future time steps. As the number of samples tends to infinity, this Monte Carlo characterization becomes an equivalent representation to the functional description of the posterior density function. The posterior filtered density at time t is approximated by

$$p(S_t | Z_{1:t}) \approx \sum_{p=1}^{N_p} w_{t,p} \delta(S_t - S_{t,p}), \quad (20)$$

where $Z_{1:t} = [Z_1 \ \cdots \ Z_m \ \cdots \ Z_t]$ are the observations and $\delta(\cdot)$ denotes the Dirac delta function. The weight $w_{t,p}$ of the particle p , is updated according to

$$w_{t,p} = w_{t-1,p} \times \frac{p(Z_t | S_{t,p}) p(S_{t,p} | S_{t-1,p})}{q(S_{t,p} | S_{t-1,p}, Z_t)}. \quad (21)$$

The choice of the importance density $q(S_{t,p} | S_{t-1,p}, Z_t)$ is an important issue in particle filter design. It can be shown that the optimal importance density (in the sense of minimizing the variance of the importance weights), conditioned upon $S_{t-1,p}$ and Z_t is $p(S_{t,p} | S_{t-1,p}, Z_t)$ [10]. However, the optimal importance density is generally not analytically available. There are other suboptimal choices. For example, a popular choice is to use the prior model density $p(S_{t,p} | S_{t-1,p})$. When substituted into (21), we obtain

$$w_{t,p} = w_{t-1,p} \times p(Z_t | S_{t,p}). \quad (22)$$

The simple and general algorithm above forms the basis of most particle filters. However, it will result in the variance of the importance weights increasing over time [10]. This will adversely affect the accuracy and lead to the degeneracy problem where, after a certain number of recursive steps, all but one particle will have negligible normalized weights. This will result in a large computational effort devoted to updating particles whose contribution to the approximation of $p(S_t | Z_{1:t})$ is almost zero. A practical measure of the degeneracy of the particle weights is the effective sample size N_{eff} introduced in [21]

$$N_{\text{eff}} = \left(\sum_{p=1}^{N_p} w_{t,p}^2 \right)^{-1}. \quad (23)$$

It is easy to see that $1 \leq N_{\text{eff}} \leq N_p$. A small N_{eff} indicates a degeneracy problem. When this occurs (e.g., when N_{eff} drops below some threshold N_{thr}), a step called resampling [9] has to be performed. Resampling eliminates sample with low weights and multiplies samples with high importance weights. This involves mapping a random measure $\{S_{t,p}, w_{t,p}\}_{p=1}^{N_p}$ into a random measure $\{S_{t,p}, 1/N_p\}_{p=1}^{N_p}$ with uniform weights.

There are several methods available when implementing the remapping step. The first introduction of resampling [9] uses random sampling of the particles based on their weights. However, a complete random selection is not necessary and it increases the Monte Carlo variation of the particles. Other methods such as stratified sampling [22] and residual sampling [23] may be applied. Systematic resampling [22] is another efficient method. It is simple to implement, it has order N_p computational complexity, and it minimizes the Monte Carlo variation.

In this paper, we make use of a particle filter to perform the inference. We use the prior $p(S_t | S_{t-1})$ as the importance function, and the systematic resampling method for resampling. We also use the effective sample size N_{eff} to determine if resampling is required at each time step.

7. Rao-Blackwellised Particle Filter (RBPF)

Rao-Blackwellisation [10, 11] in particle filters is a variance reduction technique that exploits the structure of the probabilistic model. The key idea is that the state S_t can be partitioned into two parts, $[S_t^L \ S_t^N]$, such that the posterior distribution of the nonlinear part S_t^N can be simulated using particles, and the posterior distribution of the linear part S_t^L can be updated analytically given S_t^N and some sufficient statistics. Compared to a normal particle filter that simulates the entire joint state S_t using particles, the RBPF will be more efficient, and has lower variance due to exact simulation of the conditionally linear part of the state S_t^L .

For the video tracking model using (6), (15), and (16), it is possible to Rao-Blackwellise the state space with $S_t^N = \{X_t, \theta_t\}$ and $S_t^L = \{V_t\}$.

TABLE 1: Tracking parameters for particle filter.

Algorithm parameter	Symbol	Value
Number of video frames		480
Time interval between video frames	T	1/3 s
Number of particles	N_p	200
Motion variance	q_x, q_y	$625 \text{ m}^2 \text{ s}^{-3}$
Orientation variance	q_θ	16 deg^2
Probability of becoming occluded	P_{NV}	0.1
Probability of becoming visible	P_V	0.1
Effective sample size threshold	N_{thr}	180

The derivation of the RBPF can be found in the appendix. The sufficient statistic $p(V_t | Z_{1:t}^N, S_{1:t}^N)$ is updated over time. The weights of the RBPF can then be written as

$$w_{t,p} = w_{t-1,p} \times \frac{p(Z_t | Z_{1:t-1}, S_{1:t}^N) p(S_{t,p}^N | S_{t-1,p}^N)}{q(S_{t,p}^N | S_{t-1,p}^N, Z_t)}, \quad (24)$$

where

$$\begin{aligned} p(Z_t | Z_{1:t-1}, S_{1:t}^N) &= \sum_{\nu \in \{1,0\}} p(Z_t | S_t^N, V_t = \nu) p(V_t = \nu | S_{1:t}^N, Z_{1:t-1}), \\ p(V_t | S_{1:t}^N, Z_{1:t-1}) &= \sum_{\nu \in \{1,0\}} p(V_t | V_{t-1} = \nu) p(V_{t-1} = \nu | S_{1:t-1}^N, Z_{1:t-1}). \end{aligned} \quad (25)$$

8. Simulations and Results

We applied the tracking filter to an UAV video sequence of a vehicle moving in a cluttered urban environment. The tracking parameters used are shown in Table 1. They are empirically chosen and fitted to the video data. The video data is a set of high fidelity simulations provided by General Dynamics UK Ltd. To validate the method, we manually identified the center of the vehicle as the true position in the video sequence, frame by frame.

Figure 6 shows the tracking results for the vehicle as it enters a region of occlusion due to thick smoke and then emerges. Figure 6(a) shows the RBPF Maximum A Posteriori (MAP) estimate of the vehicle's position (in green), as well as the true position of the vehicle (in magenta) in the image plane. It can be seen that the RBPF MAP estimate of the vehicle's position in the image plane is very close to the true vehicle's position. Figure 6(b) shows the corresponding MAP estimate of the vehicle's position in the real world. The drop in visibility can be seen in Figure 6(c), obtained as the posterior distribution of the visibility variable V_t . Figures 6(d) and 6(e) show the estimated orientation in the image and speed in the real world, respectively.

Figure 7 shows the distribution of the particles as the vehicle enters, passes through, and emerges from the smoke occlusion. It can be seen that the posterior distribution of the vehicle's position increases significantly as it becomes occluded from view.

We have also compared the method to a windowed Maximum Likelihood (ML) search of the template in the video sequence. The windowed ML search method proceeds by forming a window, or neighbourhood, centered about the last known position of the vehicle. For each video frame at time t , correlation scores $\text{Polar}(\tilde{x}_t, \tilde{y}_t, \theta)$ are then computed at every point in this window. Finally, the maximum score with respect to $(\tilde{x}_t, \tilde{y}_t, \theta)$ is taken as the new vehicle's location and orientation. If the score falls below a certain threshold value, the next window will be doubled in size and centered about the extrapolation of the previous two windows. This heuristic tool allows tracking to continue even when the vehicle becomes temporarily, partially, or fully occluded.

The windowed ML search is very sensitive to the threshold value. If it is set too high, then legitimate targets may be ignored. The longer this happens, the more out of date the extrapolation becomes. On the other hand, if the threshold is set too low, then, when the vehicle becomes occluded, other nearby objects will register a higher score, and the algorithm will begin to follow false positives. The ML threshold is chosen by trial and error and depends upon factors such as the noise, the template, the search area, the background, and presence of occlusions. In our experiment, a threshold of 0.6 was used in Figure 9 to successfully track the target. A threshold value of 0.5 or 0.7 lost track of the target as it entered the smoke. In this respect, the RBPF is more robust as it utilizes more optimally prior information, such as the vehicle dynamics in the real world, to predict the possible positions of the occluded vehicle.

The particle filter tracker is also more robust to multimodal behaviour of the correlation surfaces of the Polar Matching Algorithm. Figure 8 shows the particles and kernel estimate of the posterior distribution of the vehicle's position in the image plane. There are clearly two modes in the posterior distribution. The smaller mode is attributed to another vehicle. The RBPF with the tracking model is able to pick up the correct vehicle as it emerges out of the thick smoke at video frame 220. The RBPF MAP estimate is closer to the true vehicle position, compared with the windowed ML estimate.

Figure 9 shows the position error of the vehicle's MAP position estimate and the windowed ML search in the video frame when compared with the true position of the vehicle. From Figure 9, the RBPF is able to pick up the vehicle faster than the windowed ML search after it emerges from the thick smoke at video frame 220. The RBPF generally outperforms the windowed ML search slightly, with some spikes of worse performance caused by occasional errors in the ground truth data for real-world sensor positioning (a feature that is not used by the ML estimate). Further work on this topic will include modelling the random errors in sensor position measurements and this is expected to reduce or remove the spikes in the position error found for the particle filter. The ML method here used a search window of 25×25 pixels.

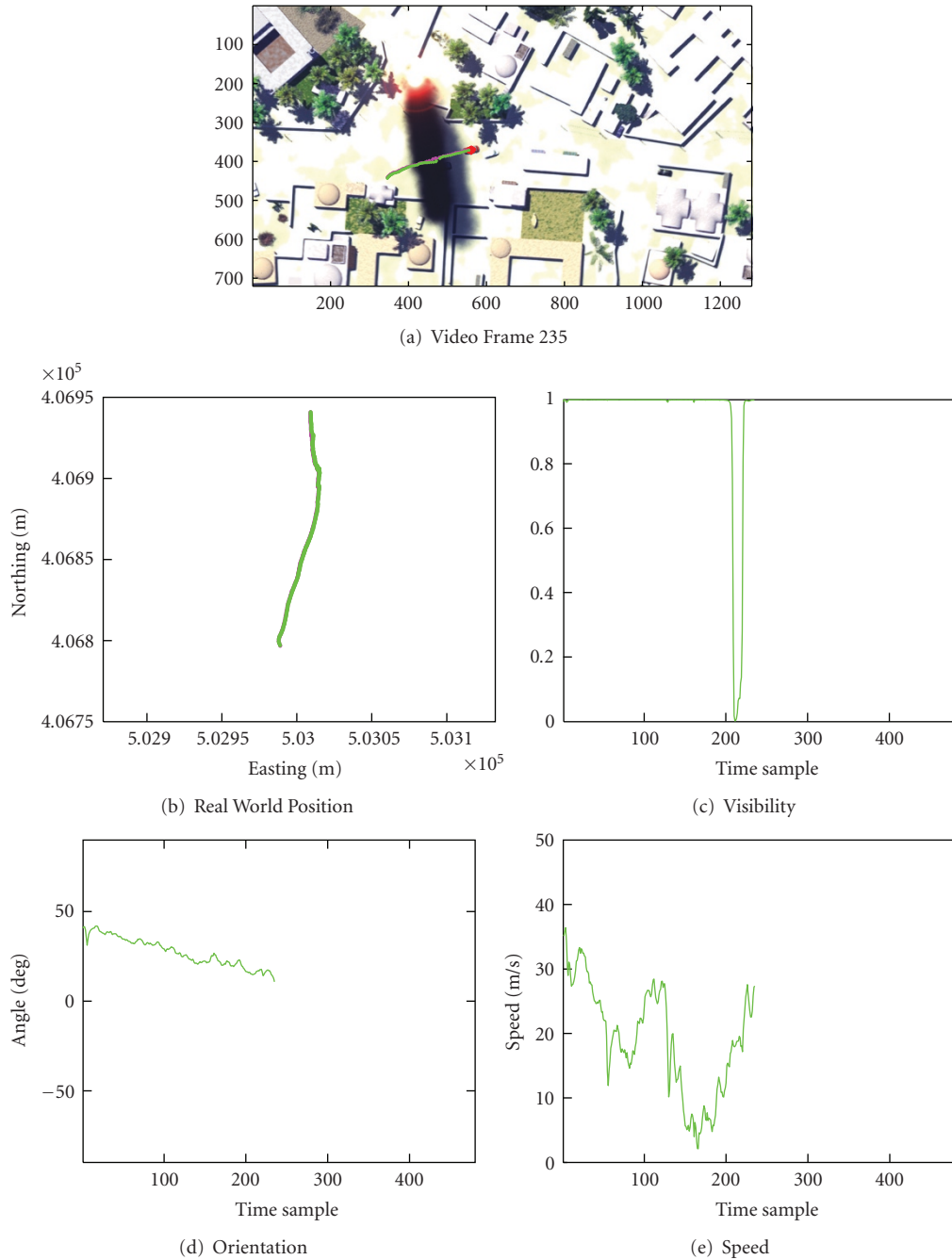


FIGURE 6: Tracking results at video frame 235. This shows the vehicle emerging from a full occlusion due to thick smoke. (a) shows the RBPf MAP estimate of the vehicle’s position (plotted in green for video frame from 175 to 235 projected onto the current video frame) and the posterior particle distribution of the position of the vehicle in the video frame (in red). The magenta track shows the true vehicle track projected onto the current video frame. It can be seen that the RBPf MAP estimate of the vehicle’s position in the image plane is very close to the true vehicle’s position. (b) shows a plot of the RBPf MAP estimate of the real world position (in green) of the vehicle from video frame 1 to 235. The magenta track shows the corresponding true vehicle track in the real world. (c), (d), and (e) show the estimated visibility, speed, and orientation of the vehicle, respectively. (c) shows clearly the decrease and subsequent increase in visibility as the vehicle enters and emerges from the smoke occlusion.

As Figure 10 illustrates, both the RBPf and windowed ML search methods use similar amount of computational resources. However, the RBPf can successfully track the vehicle in our example data with as few as 200 particles; whereas the windowed ML search method requires more

than 400 points to work without permanently losing the vehicle. The conclusion drawn here is that the particle cloud can change in size and density, according to the probabilistic model, to cover a more optimal search area than the rigid, regularly sampled, square area utilised by the windowed ML

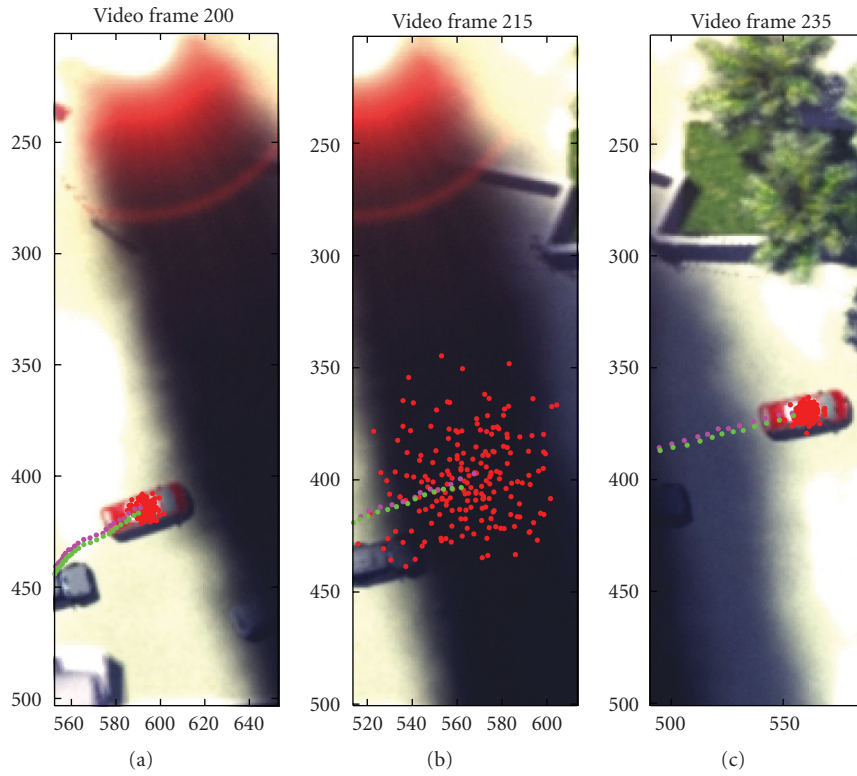


FIGURE 7: A sequential series of frames showing the distribution of particles (in red) and MAP position estimate (in green) as the vehicle enters and emerges from the thick smoke occlusion. The posterior distribution of the vehicle’s position increases significantly as it enters the occlusion. Magenta dots show the true vehicle’s path projected onto the respective video frames.

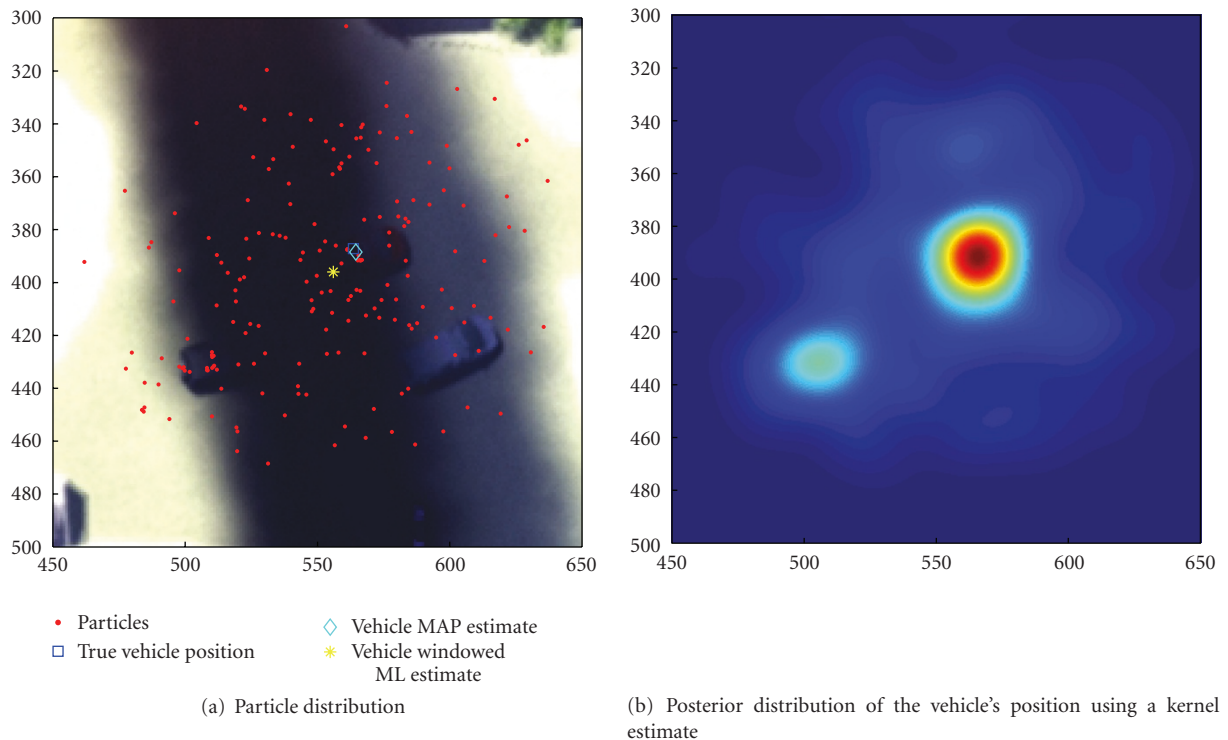


FIGURE 8: These figures show the particle distribution, as well as the corresponding kernel estimate of the posterior distribution at video frame 220.

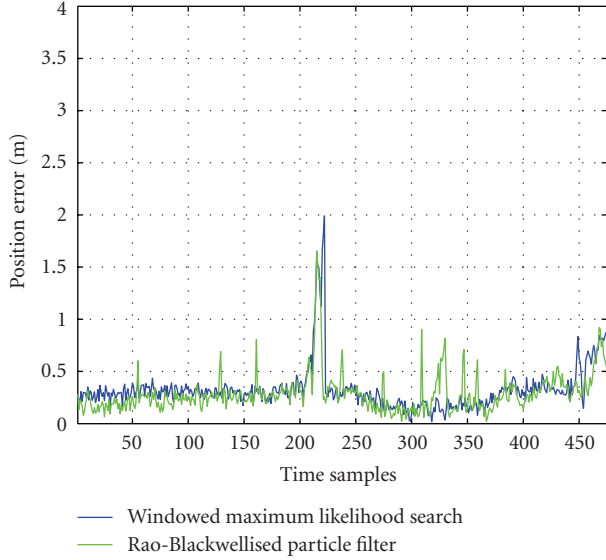


FIGURE 9: This figure shows the position errors of the vehicle's RBPF MAP position estimate (in green) and the windowed ML search estimate (in blue) with optimal threshold, when compared with the true position of the vehicle.

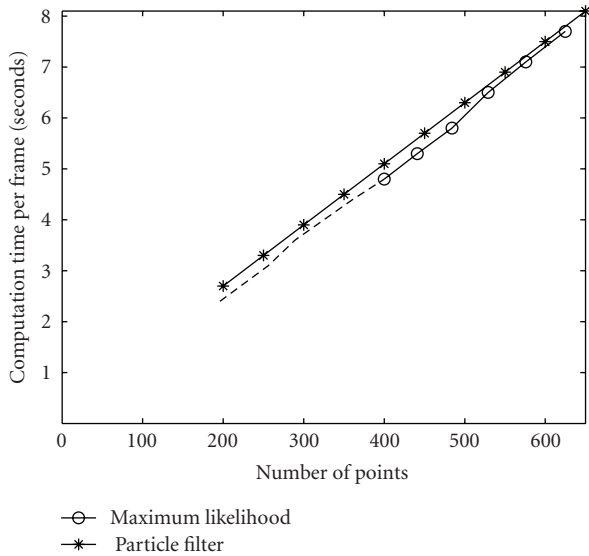


FIGURE 10: The RBPF and windowed ML search method require similar computational time per number of likelihood evaluations. However, the RBPF can work with as few as 200 particles; whereas the windowed ML search method requires more than 400 points. The dotted line denotes tracking failure. (The times were calculated by running the respective Matlab implementations on a dual core 2.4 GHz Windows machine and averaging over 10 frames.)

search approach. For the RBPF, the computational complexity is linearly proportional to the number of particles. For the windowed ML search method, the complexity is proportional to the size of the window used.

9. Conclusion

In this paper, we have shown that the combination of the rotation invariant dual-tree complex wavelet Polar Matching descriptor and the particle filter can be an effective approach to detect and track ground-based targets from UAV sensor data. Polar Matching offers target detection correlation scores for each position and orientation to the particle filter. We studied the observation statistics using the Polar Matching Algorithm and used a Track-Before-Detect likelihood to model the observation. This is further combined with a vehicle dynamical model and used in a Rao-Blackwellised Particle Filter tracking algorithm to track a vehicle in a synthetic video sequence. The RBPF provides good tracking of the vehicle and enhances the robustness of the tracking process. In the future, we will consider the case where the sensor's position and orientation are not known accurately and have to be estimated jointly with the target's position.

Appendix

Derivation of Rao-Blackwellised Particle Filter (RBPF)

We use $S_t^N = \{X_t, \theta_t\}$ to represent the general nonlinear states and $S_t^L = \{V_t\}$ for the visibility variable which we will Rao-Blackwellise in an RBPF.

In order to derive the RBPF, we first start with the joint states $\{S_{1:t}^N, V_t\}$. Then, the joint distribution given the observation $Z_{1:t}$ can be written as

$$\begin{aligned}
 & p(S_{1:t}^N, V_t | Z_{1:t}) \\
 &= (p S_t^N, S_{1:t-1}^N, V_t | Z_t, Z_{1:t-1}) \\
 &= p(Z_t | S_t^N, V_t, S_{1:t-1}^N, Z_{1:t-1}) p(S_t^N | S_{1:t-1}^N, V_t, Z_{1:t-1}) \\
 &\quad \times p(V_t | S_{1:t-1}^N, Z_{1:t-1}) p(S_{1:t-1}^N | Z_{1:t-1}) \frac{1}{p(Z_t | Z_{1:t-1})} \\
 &= p(Z_t | S_t^N, V_t) p(S_t^N | S_{1:t-1}^N) p(V_t | S_{1:t-1}^N, Z_{1:t-1}) \\
 &\quad \times p(S_{1:t-1}^N | Z_{1:t-1}) \frac{1}{p(Z_t | Z_{1:t-1})}.
 \end{aligned} \tag{A.1}$$

To Rao-Blackwellise the above distribution, we consider the marginal $p(S_{1:t}^N | Z_{1:t})$

$$\begin{aligned}
 & p(S_{1:t}^N | Z_{1:t}) \\
 &= \int p(S_{1:t}^N, V_t | Z_{1:t}) dV_t \\
 &= \left[\sum_{\nu \in \{1,0\}} p(Z_t | S_t^N, V_t = \nu) p(V_t = \nu | S_{1:t-1}^N, Z_{1:t-1}) \right]
 \end{aligned}$$

$$\begin{aligned}
& \times \frac{p(S_t^N | S_{t-1}^N) p(S_{1:t-1}^N | Z_{1:t-1})}{p(Z_t | Z_{1:t-1})} \\
& = p(Z_t | Z_{1:t-1}, S_{1:t}^N) \frac{p(S_t^N | S_{t-1}^N) p(S_{1:t-1}^N | Z_{1:t-1})}{p(Z_t | Z_{1:t-1})}. \tag{A.2}
\end{aligned}$$

The above equation gives the weight update for a particle filter, assuming we have the sufficient statistic $p(V_{t-1} | S_{1:t-1}^N, Z_{1:t-1})$. Now we need to develop a probability update from time $t-1$ to time t .

Let us consider

$$\begin{aligned}
& p(V_t | S_{1:t}^N, Z_{1:t}) \\
& = \frac{p(V_t, Z_t, Z_{1:t-1} | S_{1:t}^N)}{p(Z_{1:t} | S_{1:t}^N)} \\
& = \frac{p(Z_t | S_{1:t}^N, V_t, Z_{1:t-1}) p(V_t | S_{1:t}^N, Z_{1:t-1}) p(Z_{1:t-1} | S_{1:t}^N)}{p(Z_{1:t} | S_{1:t}^N)} \\
& = \frac{p(Z_t | S_t^N, V_t) p(V_t | S_{1:t}^N, Z_{1:t-1})}{p(Z_t | Z_{1:t-1}, S_{1:t}^N)}, \\
& p(V_t | S_{1:t}^N, Z_{1:t-1}) \\
& = \sum_{\nu \in \{1,0\}} p(V_t | V_{t-1} = \nu) p(V_{t-1} = \nu | S_{1:t-1}^N, Z_{1:t-1}) \\
& = \sum_{\nu \in \{1,0\}} p(V_t | V_{t-1} = \nu) p(V_{t-1} = \nu | S_{1:t-1}^N, Z_{1:t-1}) \tag{A.3}
\end{aligned}$$

as V_{t-1} is independent of S_t^N when given S_{t-1}^N . This can be seen by using (6) and expanding $p(V_{t-1}, S_{1:t-1}^N | S_{1:t-1}^N, Z_{1:t-1})$.

Also, we can calculate $p(Z_t | Z_{1:t-1}, S_{1:t}^N)$ using the following equation:

$$\begin{aligned}
& p(Z_t | Z_{1:t-1}, S_{1:t}^N) \\
& = \sum_{\nu \in \{1,0\}} p(Z_t | S_{1:t}^N, V_t = \nu) p(V_t = \nu | S_{1:t}^N, Z_{1:t-1}) \\
& = \sum_{\nu \in \{1,0\}} p(Z_t | S_t^N, V_t = \nu) p(V_t = \nu | S_{1:t}^N, Z_{1:t-1}). \tag{A.4}
\end{aligned}$$

From (A.3) and (A.4), we can update the sufficient statistic $p(V_t | Z_{1:t}, S_{1:t}^N)$.

We can estimate $p(V_t | Z_{1:t})$ using the particle approximation of $p(S_{1:t}^N | Z_{1:t})$

$$\begin{aligned}
p(V_t | Z_{1:t}) & = \int p(V_t | Z_{1:t}, S_{1:t}^N) p(S_{1:t}^N | Z_{1:t}) dS_{1:t}^N \\
& \approx \sum_{p=1}^{N_p} p(V_t | S_{1:t}^N, Z_{1:t}) w_{t,p} \delta(S_{1:t}^N - S_{1:t,p}^N) \\
& = \sum_{p=1}^N w_{t,p} p(V_t | S_{1:t,p}^N, Z_{1:t}). \tag{A.5}
\end{aligned}$$

Acknowledgments

This research was supported by the Data and Information Fusion Defence Technology Centre, UK, under the Fusion and Tracking Clusters. S. K. Pang's research was sponsored by Defence Science Organisation (DSO) National Laboratories of Singapore. The authors thank these parties for funding this work and would like to thank General Dynamics, UK, for providing the high fidelity synthetic data.

References

- [1] R. R. Coifman and D. Donoho, "Wavelets and statistics," in *Translation-Invariant De-Noising*, A. Antoniadis and G. Oppenheim, Eds., Lecture Notes in Statistics, pp. 125–150, Springer, Berlin, Germany, 1995.
- [2] N. Güney and A. Ertüzün, "Recognition of planar objects using multiresolution analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 70351, 9 pages, 2007.
- [3] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, 2001.
- [4] N. Kingsbury, "Rotation-invariant local feature matching with complex wavelets," in *Proceedings of the European Conference on Signal Processing (EUSIPCO '06)*, September 2006.
- [5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [6] E. P. Simoncelli and W. T. Freeman, "Steerable pyramid: a flexible architecture for multi-scale derivative computation," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, pp. 444–447, Washington, DC, USA, October 1995.
- [7] Y. Bar-Shalom and W. D. Blair, Eds., *Multitarget-Multisensor Tracking: Applications and Advances*, vol. 3, Artech House, Boston, Mass, USA, 2000.
- [8] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Boston, Mass, USA, 1999.
- [9] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian Bayesian state estimation," *IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.
- [10] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [11] O. Cappé, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899–924, 2007.

- [12] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter—Particle Filters for Tracking Applications*, Artech House, Boston, Mass, USA, 2004.
- [13] C. Kreucher, M. Morelande, K. Kastella, and A. O. Hero, “Particle filtering for multitarget detection and tracking,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, pp. 1396–1414, 2005.
- [14] S. K. Pang, J. Li, and S. J. Godsill, “Models and algorithms for detection and tracking of coordinated groups,” Tech. Rep. CUED/F-INFENG/TR-604, Department of Engineering, University of Cambridge, Cambridge, Mass, USA, July 2008.
- [15] M. Isard and J. MacCormick, “BraMBLe: a Bayesian multiple-blob tracker,” in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 34–41, January 2001.
- [16] J. Vermaak, A. Doucet, and P. Pérez, “Maintaining multimodality through mixture tracking,” in *Proceedings of the 9th IEEE International Conference on Computer Vision*, vol. 2, pp. 1110–1116, Nice, France, October 2003.
- [17] Z. Khan, T. Balch, and F. Dellaert, “MCMC-based particle filtering for tracking a variable number of interacting targets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.
- [18] S. Godsill and J. Vermaak, “Models and algorithms for tracking using trans-dimensional sequential Monte Carlo,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, pp. 976–979, May 2004.
- [19] S. J. Godsill, J. Vermaak, W. Ng, and J. Li, “Models and algorithms for tracking of manoeuvring objects using variable rate particle filters,” *Proceedings of the IEEE*, vol. 95, no. 5, pp. 925–952, 2007.
- [20] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking—part I: dynamic models,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [21] A. Kong, J. S. Liu, and W. H. Wong, “Sequential imputation and Bayesian missing data problems,” *Journal of the American Statistical Association*, vol. 89, pp. 278–288, 1994.
- [22] G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.
- [23] J. S. Liu and R. Chen, “Sequential Monte Carlo methods for dynamic systems,” *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.