Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 837405, 18 pages
doi:10.1155/2010/837405

Research Article

Robust Tracking in Aerial Imagery Based on an Ego-Motion Bayesian Model

Carlos R. del Blanco, Fernando Jaureguizar, and Narciso García

Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Correspondence should be addressed to Carlos R. del Blanco, cda@gti.ssr.upm.es

Received 23 November 2009; Revised 16 April 2010; Accepted 17 June 2010

Academic Editor: Yingzi Du

A novel strategy for object tracking in aerial imagery is presented, which is able to deal with complex situations where the camera ego-motion cannot be reliably estimated due to the aperture problem (related to low structured scenes), the strong ego-motion, and/or the presence of independent moving objects. The proposed algorithm is based on a complex modeling of the dynamic information, which simulates both the object and the camera dynamics to predict the putative object locations. In this model, the camera dynamics is probabilistically formulated as a weighted set of affine transformations that represent possible camera ego-motions. This dynamic model is used in a Particle Filter framework to distinguish the actual object location among the multiple candidates, that result from complex cluttered backgrounds, and the presence of several moving objects. The proposed strategy has been tested with the aerial FLIR AMCOM dataset, and its performance has been also compared with other tracking techniques to demonstrate its efficiency.

1. Introduction

Object tracking is a fundamental task in a wide range of military and civilian applications, such as surveillance, traffic monitoring and management, security, and defense. In applications with static cameras, the tracking process aims to locate a specific object in each frame of a video sequence using geometric, appearance, and motion features of the object. The main problem arises from the fact that there can be several location candidates for the object per frame, due to the presence of background structures, and other foreground objects similar to the target object. Furthermore, several disturbance phenomena, such as illumination changes due to weather conditions (typical in outdoor applications), variations in the object appearance because of the camera point of view, and occlusions, prevent using the criteria "the most similar candidate is the most adequate one." In order to solve this problem, additional information is used to try to recover the actual object location among the set of possible candidates. Typically, this information is the object dynamics, which is used to select the candidate location closer to the predicted location according to the equation of the object dynamics. However, a dynamic model based on the object dynamics is only valid for tracking systems with static or quasistatic cameras.

In aerial imagery applications, the camera system is mounted on a moving aerial platform, such as a plane, a helicopter, or an Unmanned Aerial Vehicle (UAV). As a consequence, the camera is not stabilized, and the acquired video sequences undergo a random global motion, called ego-motion, that prevents the use of the object dynamics to predict the future object location, making the tracking a challenging task. The ego-motion problem has been addressed in different manners in the scientific literature. They can be split into two categories: approaches based on the assumption of low ego-motion, and those based on the ego-motion estimation.

Approaches assuming low ego-motion consider that the motion component due to the camera is not very significant in comparison with the object dynamics. Under this restriction, some recent works expect that the object maintains a spatiotemporal connectivity along the sequence [1–3]; that is, the image regions related to the object in consecutive frames are spatially overlapped, and then they perform the

tracking using morphological connected operators. In cases where the hypothesis about the spatiotemporal connectivity does not hold, the most common approach is to search for the object in a bounded area centered in the location where it is expected to find the object according to its dynamics. In [4, 5] an exhaustive search is performed in a fixed-size image region, centered in the previous object location. In [6] the initial search location is estimated using a Kalman filter, and then the search is performed deterministically using the Mean Shift algorithm [7]. Other authors [8, 9] propose a stochastic search based on Particle Filtering that is able to deal with several possible location candidates, that is, local maxima/minima resulting from the cost function used to perform the search. As the displacement induced by the ego-motion increases, all these methods lose effectiveness. The reason is that the size of the search area must be larger to accommodate the expected camera ego-motion, and therefore, the probability that the tracking is distracted by false candidates dramatically increases.

On the other hand, approaches based on the ego-motion estimation are able to deal with strong ego-motion situations, in which the motion component due to the camera is quite more significant than the one corresponding to the object dynamics. Therefore, these approaches are more suitable for aerial imagery applications, in which the ego-motion causes large displacements between consecutive frames. They aim to compute the camera ego-motion between consecutive frames in order to compensate it, and thus recovering the spatiotemporal correlation of the video sequence. In airborne imagery, the scene acquired by the camera can be considered planar, since the depth relief of the objects in the scene is small enough compared to the average depth, and the field of view of the camera is also small [10]. This allows to efficiently model the camera ego-motion by a global parametric model, typically an affine or projective geometric transformation, since the effect of the parallax (apparent displacement of an object caused by a change in the location of the view point) is not significant. The existing works differ in the image registration technique used to compute the parameters of the affine or projective transformation. A thorough review of image registration techniques can be found in [11] for all kinds of vision-based applications. Another review focused on aerial imagery is presented in [12].

On the one hand, feature-based image registration techniques detect and match distinctive image features between consecutive frames to estimate a global parametric camera model. In [13], a detection and tracking system of moving objects from a moving airborne platform is described, which uses a feature-based approach to estimate an affine camera model. In [14], the KLT method is used to infer a bilinear camera model in an application that detects moving objects from a mobile robot. In the field of Forward Looking InfraRed (FLIR) imagery, the works [15–17] describe a detection and tracking system of aerial targets from an airborne platform that uses a robust statistic framework to match edge features in order to estimate an affine camera model. This system is able to successfully handle situations in which the camera motion estimation is disturbed by the presence of independent moving objects provided that there are enough detected features belonging to the background.

On the other hand, in situations in which the detection of distinctive features is particularly complicated, because the acquired images are low textured and structured, an area-based image registration technique is used to estimate the parameters of a global parametric model. In [18], a perspective camera model is computed using an optical flow algorithm for the detection of moving objects in an application of aerial visual surveillance. The optical flow algorithm is also used in [19] to estimate the parameters of a pseudo perspective camera model, which is utilized to create panoramic image mosaics. The same approach is followed in [20, 21] for a tracking application of terrestrial targets in airborne FLIR imagery. Also, for the same type of imagery, a target detection framework is presented in [22, 23], which minimizes SSDs- (Sum of Squares Differences-) based error measure to estimate an affine camera model. A similar framework of camera motion compensation is used in [24] for tracking vehicles in aerial infrared imagery, but utilizing a different minimization algorithm. In [25], the Inverse Compositional Algorithm is used to obtain the parameters of an affine camera model for a tracking application of vehicles in aerial imagery. The main problem associated with the area-based image registration techniques is that the presence of independent moving objects can drift the ego-motion estimation, especially if their sizes are significant.

Also, a combination of both feature- and area-based methods has been proposed in [26] to improve the quality of the camera compensation.

All the previous approaches, independently of the specific camera ego-motion compensation technique used, have in common that they compute only one parametric model to represent the ego-motion between consecutive frames. However, in real applications, there may be many situations where the ego-motion cannot be accurately estimated, or even where the estimation could be completely wrong, causing the tracking failure. These situations arise as a consequence of very low structured or textured scenes, where the high uncertainty, derived from the so-called aperture problem, makes almost impossible to compute the true ego-motion. Also, the presence of independent moving objects, especially if they take up large regions in the image, can drift the ego-motion estimation, since the assumption of only one global motion, that is, the ego-motion, does not hold anymore.

In this work, a novel approach for object tracking in airborne imagery undergoing strong camera ego-motion is proposed, which is able to deal with the aforementioned complex situations in order to produce a robust tracking along the time. The tracking algorithm models both the camera and object dynamics to efficiently predict the most probable object locations. The camera dynamics (i.e., the ego-motion) is probabilistically represented by a set of global parametric models, more specifically affine transformations, unlike the other approaches that only use one global parametric model. This allows to consider several possible camera ego-motions, which have the advantage to be more robust to the aforementioned aperture and independent moving

object problems. The dynamic information is combined with an appearance object model based on the detection of bright regions, which is a characteristic feature of the target objects in infrared imagery. Both appearance and dynamic models are managed by a Bayesian framework, which recursively computes the posterior probability density function (posterior pdf) of the object location. Since the resulting expression for the posterior pdf cannot be solved analytically, it is approximated by means of a Particle Filter technique [27] based on Monte Carlo simulation. Finally, an estimation of the object location is computed from the posterior pdf using a Gaussian-MMSE estimator [28], which is able to deal with situations in which the posterior pdf is clearly multimodal. In order to prove the efficiency and robustness of the proposed tracking algorithm, it has been tested on the AMCOM dataset, that is composed by a set of airborne FLIR sequences, containing many challenging tracking situations involving terrestrial vehicles. Additionally, the proposed tracking algorithm has been compared with two different tracking approaches, based also on Particle Filtering, in order to demonstrate its superior robustness and reliability.

Although the paper is focused on aerial visual tracking, the proposed tracking framework can be used in other tracking applications, provided that the scene can be considered planar; that is, the effect of the parallax is not very significant.

The rest of the paper is organized as follows. Section 2 describes the proposed tracking Bayesian filter, that combines the object appearance model, and the joint camera and object dynamic model to efficiently estimate the desired tracking information. The Particle Filtering approximation of the previous optimal, but not tractable, Bayesian filter is presented in Section 3. The estimation of the object location, based on the posterior pdf, is described in Section 4. Experimental results using the FLIR AMCOM dataset are exposed in Section 5, along with a comparison with other tracking approaches. And, lastly, the conclusions are presented in Section 6.

2. Bayesian Tracking

The tracking task is modeled by means of a Bayesian filter that aims to estimate a state vector $\mathbf{x}_k$, containing the desired tracking information, that evolves over time using a sequence of noisy observations $\mathbf{z}_{1:k} = \{\mathbf{z}_i \mid i = 1, \ldots, k\}$ up to time k. The state vector $\mathbf{x}_k = \{\mathbf{d}_k, \mathbf{g}_k\}$ contains the object dynamics (position and velocity over the image plane), $\mathbf{d}_k$, and the camera dynamics, $\mathbf{g}_k$. The observation $\mathbf{z}_k$ at time step k contains the object location candidates, which are obtained as a result of the processing of the frame I_k.

The Bayesian filter approach calculates some degree of belief in the state $\mathbf{x}_k$ at time k using the available prior information about the object and the camera and the set of observations $\mathbf{z}_{1:k}$. Therefore, the tracking problem can be formulated as the estimation of the posterior probability density function (posterior pdf) of the state of the object, $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$, which is recursively calculated by means of two stages: prediction and update. The prediction stage involves to obtain the prior pdf of the state $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ at time k via the Chapman-Kolmogorov equation:

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) &= \int p(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \\ &= \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}, \end{aligned} \tag{1}$$

where $p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1})$ is the posterior pdf at the previous time step, and $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ is the state transition probability, that encodes the information about the object and camera dynamics. The object dynamics is modeled by the linear function:

$$\mathbf{d}_k = \mathbf{M} \cdot \mathbf{d}_{k-1}, \tag{2}$$

where $\mathbf{M}$ is a matrix that represents a first-order linear system of constant velocity. This object dynamic model is a reasonable approximation for a wide range of object tracking applications, provided that the camera frame rate is enough high. The camera dynamics is modeled by an affine geometric transformation $\mathbf{g}_k$, which is a satisfactory approximation of the ideal projective camera model for the case of aerial imagery, since the depth relief of the objects in the scene is small enough compared to the average depth, and the field of view is also small [10]. Then, combining both models, the joint object and camera dynamics can be expressed as

$$\mathbf{d}_k = \mathbf{g}_k \cdot \mathbf{M} \cdot \mathbf{d}_{k-1}, \tag{3}$$

which, firstly, predicts the object position and velocity according to the object dynamic model, and then, it rectifies them using the affine transformation to compensate the camera motion.

Based on this joint dynamic model, the transition probability $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ can be expressed as

$$\begin{aligned} p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) &= p(\mathbf{d}_k, \mathbf{g}_k \mid \mathbf{d}_{k-1}, \mathbf{g}_{k-1}) \\ &= p(\mathbf{d}_k \mid \mathbf{d}_{k-1}, \mathbf{g}_{k-1:k}) p(\mathbf{g}_k \mid \mathbf{d}_{k-1}, \mathbf{g}_{k-1}) \\ &= p(\mathbf{d}_k \mid \mathbf{d}_{k-1}, \mathbf{g}_k) p(\mathbf{g}_k), \end{aligned} \tag{4}$$

where it has been assumed that, on the one hand, the current object position is conditionally independent of the camera motion in the previous time step (as the proposed joint dynamic model states), and, on the other hand, the current camera motion is conditionally independent of both the camera motion and the object position in previous time steps. This last assumption results from the fact that the camera ego-motion is completely random, not following any specific pattern. The probability term $p(\mathbf{d}_k \mid \mathbf{d}_{k-1}, \mathbf{g}_k)$ models the uncertainty of the proposed joint dynamic model as

$$p(\mathbf{d}_k \mid \mathbf{d}_{k-1}, \mathbf{g}_k) = N(\mathbf{d}_k; \mathbf{g}_k \cdot \mathbf{M} \cdot \mathbf{d}_{k-1}, \sigma_{\text{tr}}^2), \tag{5}$$

where $N(x; \mu, \sigma^2)$ is a Gaussian or Normal distribution of mean μ and variance σ^2. Thus, the term σ_{tr}^2 represents the unknown disturbances of the joint dynamic model.

The other probability term in (4), $p(\mathbf{g}_k)$, expresses the probability that one specific geometric transformation

represents the true camera ego-motion between consecutive time steps. For the ongoing tracking application, dealing with infrared imagery, the probability of a specific geometric transformation $\mathbf{g}_k$ is based on the quality of the image alignment achieved by $\mathbf{g}_k$ between consecutive frames. The quality of the image alignment is computed by means of the Mean Square Error function, $\text{mse}(x, y)$, between the current frame $\mathbf{I}_k$, and the previous frame $\mathbf{I}_{k-1}$ warped by the transformation $\mathbf{g}_k$. Thus, the probability $p(\mathbf{g}_k)$ is mathematically expressed as

$$p(\mathbf{g}_k) = N\left(\text{mse}(\mathbf{I}_k, \mathbf{g}_k \cdot \mathbf{I}_{k-1}); 0, \sigma_g^2\right), \tag{6}$$

where $N(x; \mu, \sigma^2)$ is a Gaussian distribution of mean μ and variance σ^2, and σ_g^2 is the expected variance of the image alignment process.

After the prediction stage, the update stage aims to reduce the uncertainty of the predicted $p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})$ using the new available observation $\mathbf{z}_k$ (observations are available at discrete times) through Bayes' rule:

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k)p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})}, \tag{7}$$

where $p(\mathbf{z}_k \mid \mathbf{x}_k)$ is the likelihood function that evaluates the degree of support of the observation $\mathbf{z}_k$ to the predicted $\mathbf{x}_k$. Finding an observation model for the likelihood $p(\mathbf{z}_k \mid \mathbf{x}_k)$ in airborne infrared imagery, that appropriately describes the object appearance and its variations along the time, is quite challenging due to the special characteristics of the infrared imagery (low signal-to-noise ratio, target objects low contrasted with the background, and nonrepeatability of the target signature), changes in illumination, variations in the 3D viewpoint, and changes in the object size along the sequence. The most robust and reliable object property is the presence of bright regions or, at least, regions that are brighter than their surrounding neighborhood, which typically correspond to the engine and exhaust areas of the object. Based on this fact, the likelihood function uses an observation model that aims to detect the main bright regions of the target. This is accomplished by a rotationally symmetric Laplacian of Gaussian (LoG) filter, characterized by a sigma parameter that is tuned to the lowest dimension of the object size, so that the filter response is maximum in the bright regions with a size similar to the tracked object. The main handicap of the observation model is its lack of distinctiveness, since whatever bright region with an adequate size can be the target object. As consequence, the resulting LoG filter response is strongly multimodal. This fact, coupled with the camera ego-motion, dramatically complicates a reliable estimation of the state vector. This situation is illustrated in Figures 1 and 2. The first one, Figure 1, shows two consecutive frames, (a) and (b), of an infrared sequence acquired by an airborne camera, in which the target object has been enclosed by a rectangle. Figure 2 shows the LoG filter response related to Figure 1(b), where the own image has been projected over the filter response for a better interpretation, in such a way that the upper left corner of Figure 1(b) corresponds with the origin of coordinates of Figure 2. The multimodality feature is clearly observed, and in theory any of the modes could be the right object position. Moreover, for this specific case, if only the object dynamics is considered, the closest mode to the predicted object location (marked by a vertical black line) is not the true object location, because of the effects of the camera ego-motion.

Based on the previous observation model, and assuming that $\mathbf{z}_k$ is conditionally independent of $\mathbf{g}_k$ given $\mathbf{d}_k$, the likelihood probability can be expressed as

$$\begin{aligned} p(\mathbf{z}_k \mid \mathbf{x}_k) &= p(\mathbf{z}_k \mid \mathbf{d}_k, \mathbf{g}_k) \\ &= p(\mathbf{z}_k \mid \mathbf{d}_k) = N(\mathbf{z}_k; \mathbf{d}_k, \sigma_L^2), \end{aligned} \tag{8}$$

where $\mathbf{z}_k$ is the LoG filter response of the frame $\mathbf{I}_k$, and the variance σ_L is set to highlight the main modes of $\mathbf{z}_k$, while discarding the low significant ones. This is illustrated in Figure 3, where only the most significant modes of Figure 2 are highlighted.

The denominator of (7) is just a normalizing constant given by

$$\begin{aligned} p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1}) &= \int p(\mathbf{z}_k, \mathbf{x}_k \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_k \\ &= \int p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_k. \end{aligned} \tag{9}$$

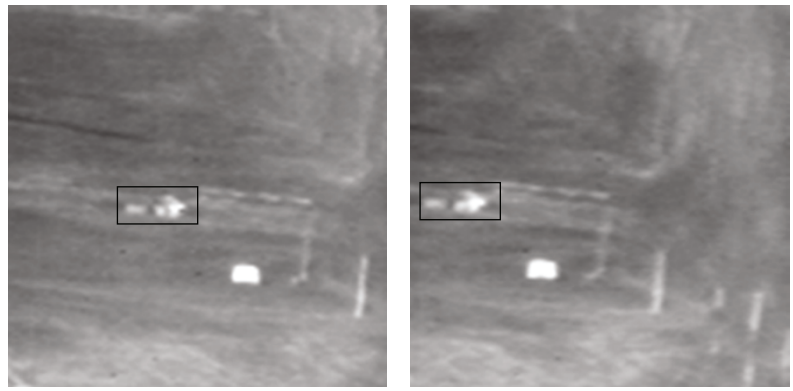
The initial pdf $p(\mathbf{x}_0 \mid \mathbf{z}_0) \equiv p(\mathbf{x}_0)$, called the prior, is initialized as a Kronecker's delta function $\delta(\mathbf{x}_0)$ using the ground truth information. In a general case, $p(\mathbf{x}_0)$ could be initialized as a Gaussian function using the information given by an object detector algorithm, as in [1, 2, 15–17, 22, 23].

In practice, the computation of the posterior pdf, by means of the recursive (1) and (7), is not feasible, since the dynamic and observation models are nonlinear and non-Gaussian. As a result, the use of approximate inference methods is necessary. In the next section, a Particle Filtering strategy is presented to obtain an approximate solution of the posterior pdf.

3. Particle Filter Approximation

The optimal solution of the posterior pdf $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$, given by (7), cannot be determined analytically in practice, but it can be approximated using suboptimal methods. Particle Filtering is an approximate inference method based on Monte Carlo simulation for solving Bayesian filters. In contrast to other approximate inference methods, such as Extended Kalman Filters, Unscented Kalman Filters, and Hidden Markov Models, Particle Filtering is able to deal with continuous state spaces and nonlinear/non-Gaussian processes [29], conditions that arise in real tracking situations. The Particle Filter technique approximates $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ by a set of N_S-weighted random samples $\{\mathbf{x}_k^i, i = 1, \ldots, N_S\}$ [27]:

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \frac{1}{c} \sum_{i=1}^{N_S} w_k^i \delta\left(\mathbf{x}_k - \mathbf{x}_k^i\right), \tag{10}$$



(a) (b)

Figure 1: Two consecutive frames of an FLIR sequence acquired by an airborne camera.

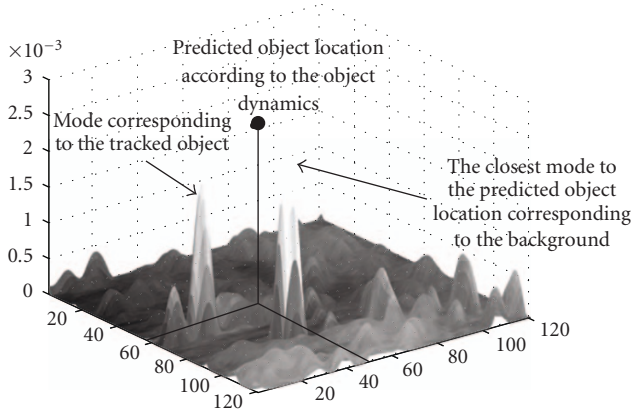


Figure 2: Multimodal LoG filter response related to Figure 1(b).

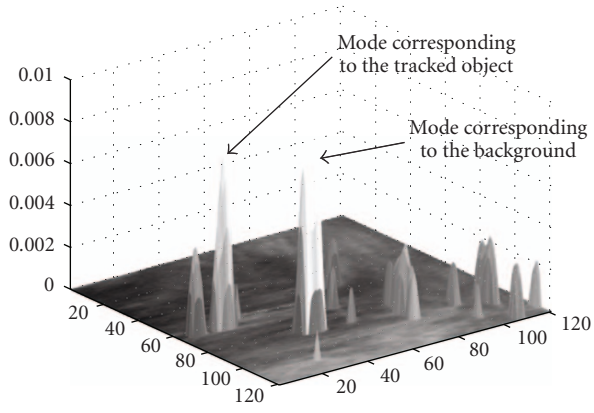


Figure 3: Likelihood distribution related to Figure 2.

where the function $\delta(x)$ is Kronecker's delta, $\{w_k^i, i = 1, \ldots, N_S\}$ is the set of weights related to the samples, and $c = \sum_{i=1}^{N_S} w_k^i$ is a normalization factor. As the number of samples becomes very large, this approximation becomes equivalent to the true posterior pdf.

Both samples $\mathbf{x}_k^i$ and weights w_k^i are obtained using the concept of importance sampling [27, 28], which aims to reduce the variance of the estimation given by (10) by means of a Monte Carlo simulation. The set of samples $\{x_k^i, i = 1, \ldots, N_S\}$ is drawn from a proposal distribution function $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)$, called the importance density. The optimal $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)$ should be proportional to $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ and should have the same support (the support of a function is the set of points where the function is not zero), since in this case the variance is zero. But this is only a theoretical solution, since it would imply that $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ is known. The approach followed in this paper is to approximate the importance density by the likelihood and the prior probability of the camera motion:

$$q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k) = p(\mathbf{z}_k \mid \mathbf{d}_k) p(\mathbf{g}_k), \tag{11}$$

which is an efficient simplification of the optimal, but not tractable, importance density $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)$ [29].

The samples $\mathbf{x}_k^i = \{\mathbf{d}_k^i, \mathbf{g}_k^i\}$ are drawn from the previous proposal distribution by a hierarchical sampling strategy. This, firstly, draws samples $\mathbf{g}_k^i$ from $p(\mathbf{g}_k)$ and then draws samples $\mathbf{d}_k^i$ from $p(\mathbf{z}_k \mid \mathbf{d}_k)$.

The sampling procedure for obtaining samples $\mathbf{g}_k^i$ from $p(\mathbf{g}_k)$ is based on a two-stage strategy, that firstly performs a fast, but rough, sampling of the affine space, and lastly improves the affine sampling by refining the samples with higher probability through a more expensive and accurate procedure. This two stage strategy allows to efficiently obtain a probabilistic representation of the camera motion with a relatively low computational cost. Section 3.1 describes the sampling procedure in more detail.

The object dynamic samples $\mathbf{d}_k^i$ are drawn from the likelihood $p(\mathbf{z}_k \mid \mathbf{d}_k)$ (11), which is a convenient decision since the main modes of the posterior distribution also appear in the likelihood function. Sampling from the likelihood function is not a trivial task, since it is a bivariate function composed by narrow modes (see Figure 3). To deal with this issue, a Markov Chain Monte Carlo (MCMC) sampling method is proposed, which is able to efficiently represent the likelihood function by a reduced number of samples. Section 3.2 describes the MCMC sampling procedure in more detail.

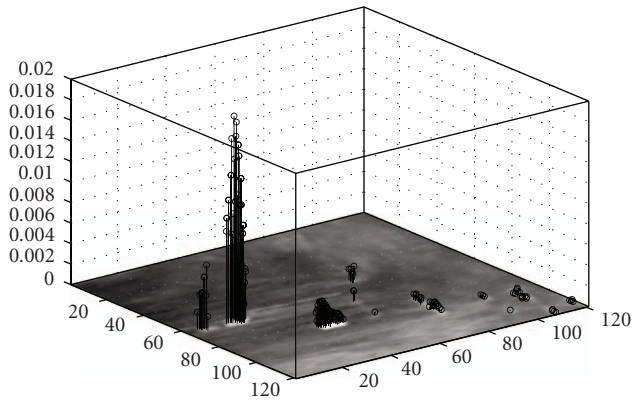


Figure 4: Particle Filtering-based approximation of the posterior probability $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$.

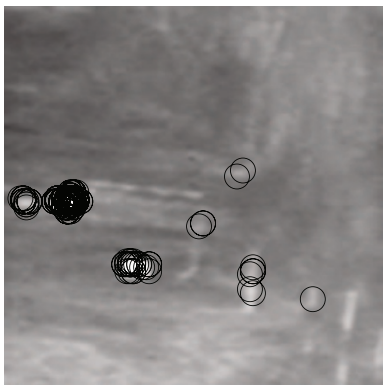


Figure 5: SIR resampling of $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$.

Once that the samples $\mathbf{x}_k^i = \{\mathbf{d}_k^i, \mathbf{g}_k^i\}$ have been obtained, the weights w_k^i are computed by [29]

$$w_k^i = w_{k-1}^i \frac{p\left(\mathbf{z}_k \mid \mathbf{x}_k^i\right) p\left(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i\right)}{q\left(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k\right)}. \tag{12}$$

Using the likelihood, transition, and importance density probabilities, this expression can be simplified as

$$\begin{aligned} w_k^i &= w_{k-1}^{\mathrm{i}} \frac{p\left(\mathbf{z}_k \mid \mathbf{d}_k^i\right) p\left(\mathbf{d}_k^i \mid \mathbf{d}_{k-1}^i, \mathbf{g}_k^i\right) p\left(\mathbf{g}_k^i\right)}{p\left(\mathbf{z}_k \mid \mathbf{d}_k^i\right) p\left(\mathbf{g}_k^i\right)} \\ &= w_{k-1}^i p\left(\mathbf{d}_k^i \mid \mathbf{d}_{k-1}^i, \mathbf{g}_k^i\right). \end{aligned} \tag{13}$$

According to this expression, the samples that best fit with the joint camera and object dynamic model will have more relevance than the rest.

The importance sampling principle has a serious drawback, called the degeneracy problem [27], consisting in only one weight has a significant value after a few iterations, while the rest of weights has an inconsiderable value. In order to overcome this problem, a resampling step is applied to reduce the degeneracy problem. This is accomplished by means of the Sampling Importance Resampling (SIR) algorithm that selects more times the samples with higher weights, while the ones with an insignificant weight are discarded. After SIR resampling, all the samples have the same weight.

Figures 4 and 5 show the estimated posterior probability, $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$, and the result of applying the SIR resampling, respectively. Notice that the samples corresponding with modes related to background structures have a lower weight than the ones related to the tracked object, due to the coherence with the expected camera and object dynamics. As a result, the estimated posterior pdf concentrates all the meaningful samples in the target object region.

3.1. Sampling of the Affine Space. The sampling procedure for obtaining samples $\mathbf{g}_k^i$ from $p(\mathbf{g}_k)$ is based on a two-stage strategy, that firstly draws a set of affine transformation samples that represent a rough estimation of $p(\mathbf{g}_k)$ and then refines the sampling by improving the accuracy of the samples with higher weight using a complex algorithm.

The goal of the first stage is to compute with a low computational cost a set of affine transformation samples, which represent a rough approximation of the underlying $p(\mathbf{g}_k)$. The algorithm is based on a fast uniform sampling that uses the available prior knowledge for bounding the range of possible affine parameters and for estimating an appropriate sampling step. For the purpose of bounding the range of affine parameters, a subset of the video sequences used to test the proposed tracking algorithm have been used as training set, in order to analyze the set of the expected camera motions. These sequences belong to the infrared AMCOM dataset (see Section 5) and have been acquired by different infrared cameras on board an aerial platform. The camera motion estimation in this training set has been supervised by a user to accurately and reliably obtain the actual camera motion. The resulting analysis reveals that the most significant motions are translations, which can reach a value close to the half of the image size for some extreme situations. On the contrary, the magnitude of the scale, rotation, and shear transformations is much less significant, close to identity matrix transformation. On the other hand, the choice of the sampling step depends on the capability of the whole sampling procedure to converge to the actual affine transformation given an initial affine transformation sample. Regarding the convergence, the sampling step should be small to ensure that at least the distance in the affine space between one sample and the actual affine transformation that represent the camera motion is short enough. But considering the computational cost, the sampling step should be as large as possible. The convergence capability has been experimentally measured by synthetically warping an image by different affine transformations of increasing magnitude, until the converge to the actual camera motion is not possible. In addition, since the convergence capability depends on the scene structure, this process has been performed with a set of different images belonging to several sequences of the AMCOM dataset. As a result, the sampling step for the translation components must be less than 8 pixels, while for the rest of motion components a unique sample is enough, which assumes no

scale, rotation, and/or shear distortion, since the sampling procedure satisfactorily achieves the convergence to the real affine parameters for camera motions that take place in the AMCOM video sequences. Taking into account the previous sampling guidelines, the initial set of affine transformation samples has the form

$$\mathbf{t}_k^i = \begin{bmatrix} 1 & 0 & \left(t_x\right)_k^i \\ 0 & 1 & \left(t_y\right)_k^i \\ 0 & 0 & 1 \end{bmatrix}, \quad i = 1, \ldots, N_S, \tag{14}$$

where $(t_x)_k^i$ and $(t_y)_k^i$ are the translation components, with a sampling step less than 8 pixels. For the ongoing tracking application, the sampling step has been fixed to 5, which is a good tradeoff between accuracy and computational cost. Note that the rest of affine parameters of $\mathbf{t}_k^i$ are equivalent to the identity matrix, meaning that there is no scale, rotation, and shear warping with respect to the previous image frame, since, as stated before, the whole sampling procedure can satisfactorily deal with these kinds of distortions in the AMCOM dataset. Figure 6 shows the initial set of affine transformations, $\{\mathbf{t}_k^i, i = 1, \ldots, 441\}$, arranged in a 21×21 grid.

The set of initial affine transformations $\{\mathbf{t}_k^i \mid i = 1, \ldots, N_S\}$ are evaluated by checking the consistency of the scene structure between the current image and the compensated one, that is, the previous image warped by the affine transformation sample under evaluation. Two images have a similar scene structure when their image edges have a similar shape and spatial arrangement, indicating that they are closely aligned. The scene structure of an image is characterized by a set of shape descriptors, called extended shape contexts (E-SCs). The shape context descriptor was originally proposed by Belongie et al. [30] for recognizing 2D and 3D objects in low clutter situations. Mori and Malik [31] proposed an extended version of the shape context, the E-SC, to achieve a greater robustness to the clutter. The first step to evaluate the consistency of the scene structure between the previous image warped by the affine transformation under evaluation, $\mathbf{t}_k^i \cdot I_{k-1}$, and the current image, I_k, consists in computing the most relevant edges of both images using the Canny algorithm. A uniform random sampling of the edge locations of I_k is carried out, and then an E-SC descriptor is computed in each sampled location. Both the set of E-SC descriptors and their spatial distribution define the scene structure. Another set of E-SC descriptors is computed using the detected edges in I_{k-1}. The locations of the E-SC descriptors are the same as those of I_k, but warped by the transformation $\mathbf{t}_k^i$ under evaluation. This approach is computationally much more efficient than warping the whole image I_{k-1} using $\mathbf{t}_k^i$. The similarity of both sets of descriptors is measured by computing the Bhattacharyya distance between corresponding E-SC descriptors. The consistency of the scene structure is then obtained by summing the contributions of all the distances. A low value of the consistency of the scene structure means that I_k and $\mathbf{t}_k^i \cdot I_{k-1}$ are roughly aligned. The samples $\{\mathbf{t}_k^i \mid i = 1, \ldots, N_S\}$ and their associated weights, given by the values of consistency of the scene structure, are a rough estimation of $p(\mathbf{g}_k)$.

Figure 7 shows the weights of the affine transformations $\mathbf{t}_k^i$ used to roughly approximate the camera motion probability $p(\mathbf{g}_k)$ between two consecutive time steps. The weights are arranged in the same way of the previous grid of initial transformations and are encoded with a color scale. In this case, the maximum weight corresponds with $\mathbf{t}_k^{74}$.

The second stage refines the previous rough estimation of $p(\mathbf{g}_k)$ by means of an image registration algorithm presented in [32]. This method assumes an initial geometric transformation $\mathbf{t}_k^i$ and then uses the whole image intensity information to compute a global affine transformation $\mathbf{g}_k^i$, which is an improved estimation of the camera motion. This method explicitly accounts for global variations in image intensities to be robust to illumination changes. To reduce the computational cost, only the samples $\mathbf{t}_k^i$ with higher probability are used to improve the estimation of $p(\mathbf{g}_k)$. Finally, the set of affine transformations $\{\mathbf{g}_k^i \mid i = 1, \ldots, N_S\}$ is obtained by means of an SIR resampling, which makes a random selection of the affine transformation samples according to their weights. The resulting set of affine transformations is an accurate approximation of the underlying camera motion probability.

An alternative approach to the SIR resampling could be to select the sample with the highest weight, since it should represent the most accurate camera motion. In this case, the sampling procedure would be equivalent to an optimization approach based on an stochastic search, since only the best sample is used. However, the statement "the highest $p(\mathbf{g}_k^i)$ corresponds with the most accurate camera motion estimation" is not always true. For example, in situations with independent moving objects, the camera ego-motion estimation can be biased by the moving objects. Also, a poor estimation is obtained when the effects of the aperture problem [33, 34] are quite significant. As a consequence, in both situations the actual camera motion could be represented by one $\mathbf{g}_k^i$ with a probability value lower than the one with the maximum probability value. For this reason, a probabilistic representation of the camera motion based on discrete samples is more efficient than a deterministic approach that estimates the best transformation.

3.2. MCMC Sampling of the Likelihood Function. The object dynamic samples $\mathbf{d}_k^i$ are drawn from the likelihood $p(\mathbf{z}_k \mid \mathbf{d}_k)$ (11) to finally obtain $\mathbf{x}_k^i = \{\mathbf{d}_k^i, \mathbf{g}_k^i\}$. This is a convenient decision since the main modes of the posterior distribution also appear in the likelihood function. Sampling from the likelihood function is not a trivial task, since it is a bivariate function composed by narrow modes (see Figure 3). To deal with this issue, a Markov Chain Monte Carlo (MCMC) sampling method is proposed, which is able to efficiently represent the likelihood function by a reduced number of samples. The MCMC approach generates a sequence of samples $\{\mathbf{d}_k^i, i = 1, \ldots, N_S\}$ by means of a Markov Chain, in such a way that the stationary distribution is exactly the target distribution. The Metropolis-Hasting [28, 35] algorithm is an MCMC method that uses a proposal distribution for simulating such a chain. The appropriate

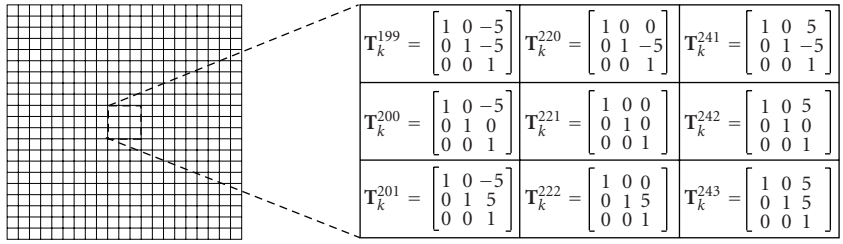


Figure 6: Initial set of affine transformation samples used to roughly approximate $p(\mathbf{g}_k)$.

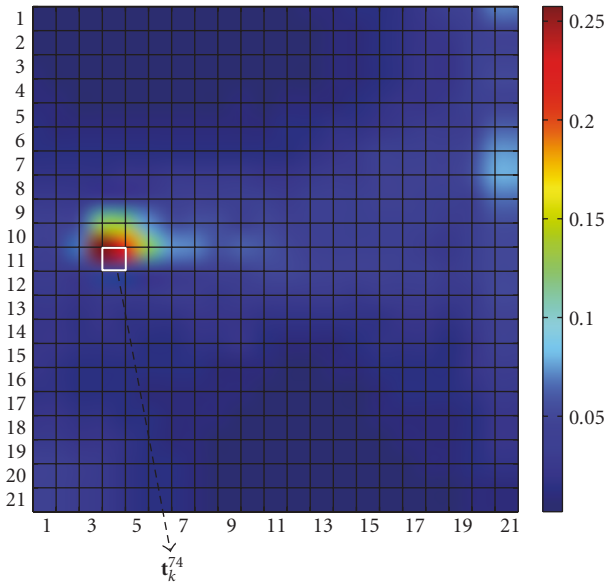


Figure 7: Weights of $\{\mathbf{t}_k^i, i = 1, \ldots, 441\}$, which roughly approximate $p(\mathbf{g}_k)$.

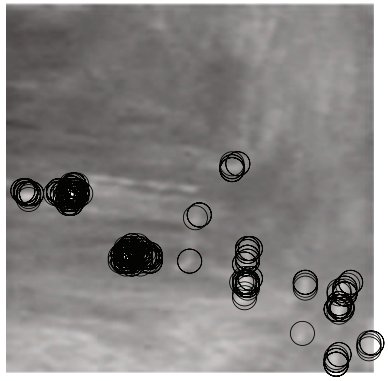


Figure 8: Metropolis-Hasting sampling of the likelihood distribution depicted in Figure 2.

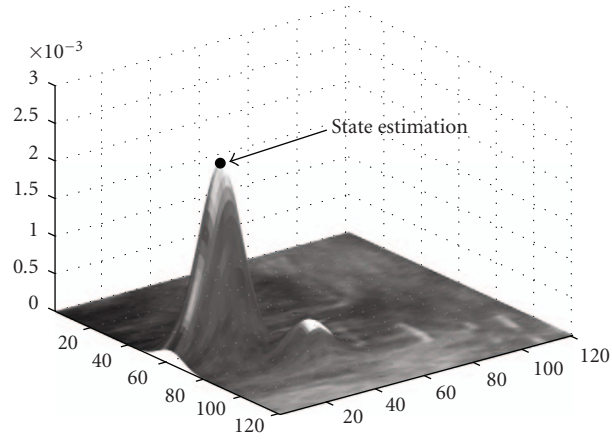
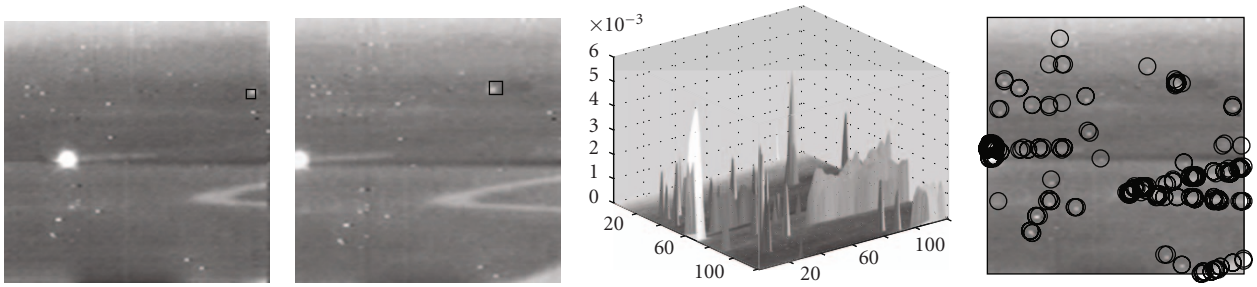


Figure 9: Result of applying the Gaussian kernel over $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ (depicted in Figure 4), along with the final state estimation $\hat{\mathbf{x}}_k$ marked by a black circle.

Figure 10: Tracked object accurately enclosed by a white rectangle.

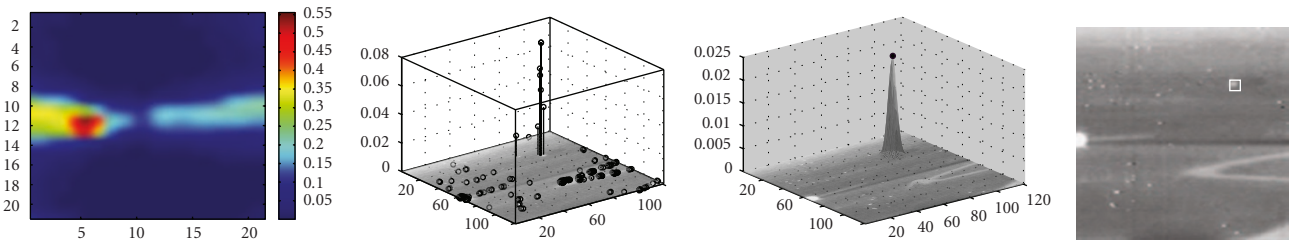
selection of the proposal distribution is the key for the efficient sampling of the target distribution. For the case of the likelihood $p(\mathbf{z}_k \mid \mathbf{d}_k)$ sampling, a Gaussian function, with mean zero and a variance proportional to the lowest size dimension of the tracked object, has proven to be efficient. Another fundamental issue is the initialization of the Markov Chain. Since the likelihood function concentrates almost all the probability in a few sparse regions of the state space (i.e., in its sparse narrow modes), the Markov Chain needs a large amount of samples to correctly simulate it. A more efficient approach is to use a set of Markov Chains, with different initialization states given by the main local maxima of the likelihood distribution. In this way, the likelihood is efficiently simulated by a reduced number of samples located on the main modes.

Figure 8 shows the result of applying the proposed Metropolis-Hasting sampling algorithm to simulate the



(a) Frame 15 (b) Frame 16 (c) Likelihood $p(\mathbf{z}_k \mid \mathbf{x}_k)$ (d) Sampling

Figure 11: Common intermediate results for all the three tracking algorithms in a situation of strong ego-motion.



(a) Probability values of $\mathbf{g}_k^i$ before SIR (b) Sampled posterior probability (c) Gaussian-MMSE-based estimation (d) Object tracking result

Figure 12: Tracking results for the BEH algorithm in a situation of strong ego-motion.

likelihood distribution depicted in Figure 3. The samples have been marked with circles. Notice that the samples are on the main modes of the likelihood distribution, in spite of the relatively low number of used samples.

4. State Estimation

The estimated posterior pdf, $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$, embodies all the available statistical information, allowing the computation of an optimal estimation of the state of the object $\hat{\mathbf{x}}_k$. In general terms, the resulting posterior probability can be quasi-unimodal (if there is only one significant mode) or multimodal. This fact depends on the distance between the mode corresponding to tracked object and the modes relative to the background in the likelihood function. While for the case of a quasi-unimodal posterior probability, the state estimation can be efficiently performed by means of the MMSE estimator, for the case of a multimodal posterior probability, the MMSE estimator does not produce a satisfactory estimation, since the background modes bias the result. To avoid such a bias in the estimation, the MMSE estimator should only use the samples relative to the tracked object mode, discarding the rest. This is achieved by means of a bivariate Gaussian kernel $N(\mathbf{x}; \mu_e, \Sigma_e)$ of mean μ_e and covariance matrix Σ_e [28], which gives more relevance to the samples located close to the Gaussian mean. In this way, when the Gaussian mean is centered over the tracked object mode, only the samples related to this mode will have a significant value. The proposed Gaussian-MMSE estimator is mathematically expressed as

$$\hat{\mathbf{x}}_k = \max\left(\frac{1}{N_S}\sum_{l=1}^{N_S}\sum_{i=1}^{N_S} N\left(\mathbf{x}_k^i; \mathbf{x}_k^l, \Sigma_e\right) p\left(\mathbf{x}_k^i \mid \mathbf{z}_{1:k}\right)\right), \quad (15)$$

where the covariance matrix Σ_e determines the bandwidth of the Gaussian kernel, which must be coherent with the size of the tracked object mode. Taking into account the relationship between the size of the tracked object mode and the bandwidth of the LoG filter used in the object detection (Section 2), that in turn it was set according to the object size, an efficient covariance matrix can be estimated as

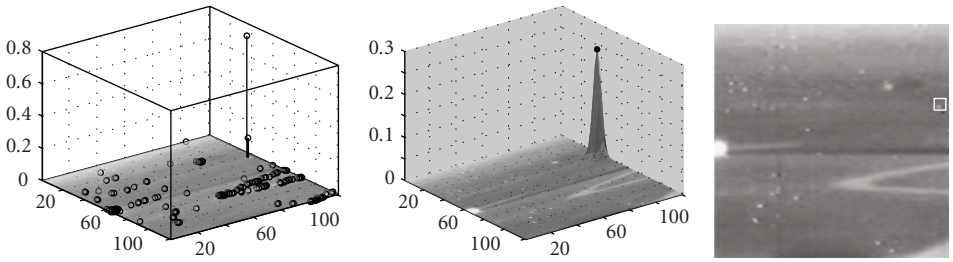
$$\Sigma_e = \begin{bmatrix} \frac{s_x}{2} & 0 \\ 0 & \frac{s_y}{2} \end{bmatrix}, \quad (16)$$

where s_x and s_y are the width and height of the object, respectively, which are the same parameters as the ones used in the LoG-based object detector.

Figure 9 shows the result of applying the Gaussian kernel over $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$, along with the maximum corresponding to the final estimation $\hat{\mathbf{x}}_k$, that has been marked by a black circle. Figure 10 shows the tracked object accurately enclosed by a white rectangle corresponding to the estimated $\hat{\mathbf{x}}_k$.

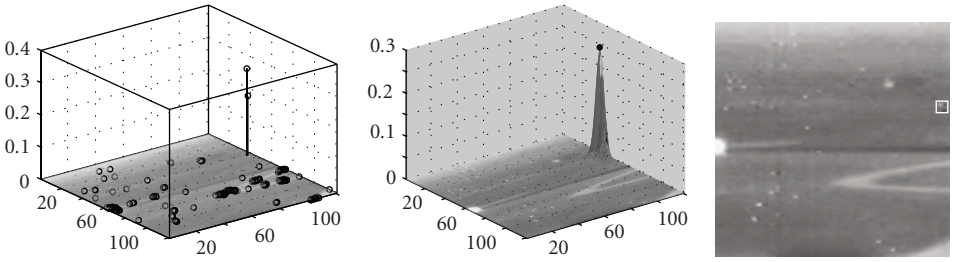
5. Results

The proposed object tracking algorithm has been tested using the AMCOM dataset. This consists of 40 infrared



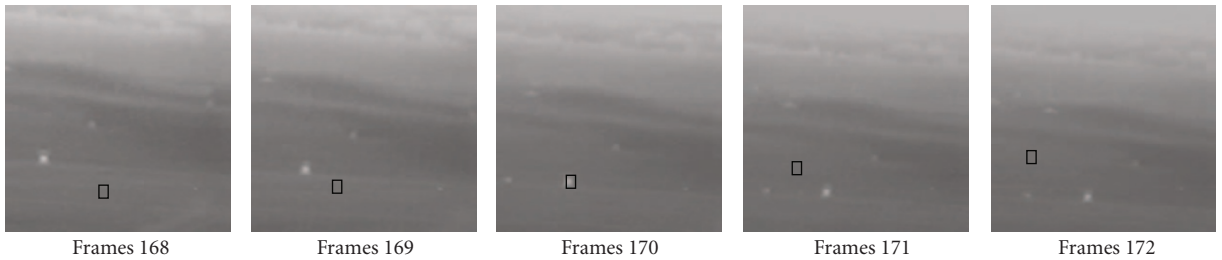
(a) Sampled posterior probability (b) Gaussian-MMSE based estimation (c) Object tracking result

Figure 13: Tracking results for the DEH algorithm in a situation of strong ego-motion.



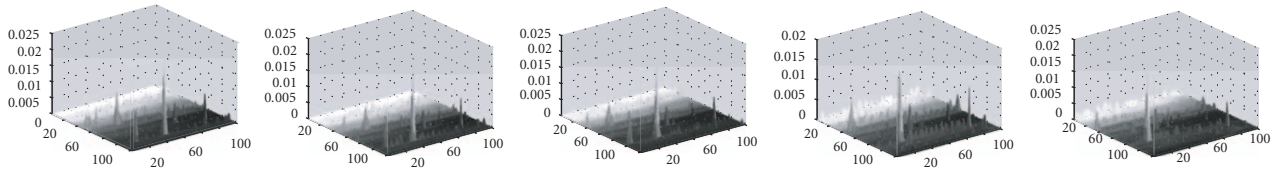
(a) Sampled posterior probability (b) Gaussian-MMSE based estimation (c) Object tracking result

Figure 14: Tracking results for the NEH algorithm in a situation of strong ego-motion.



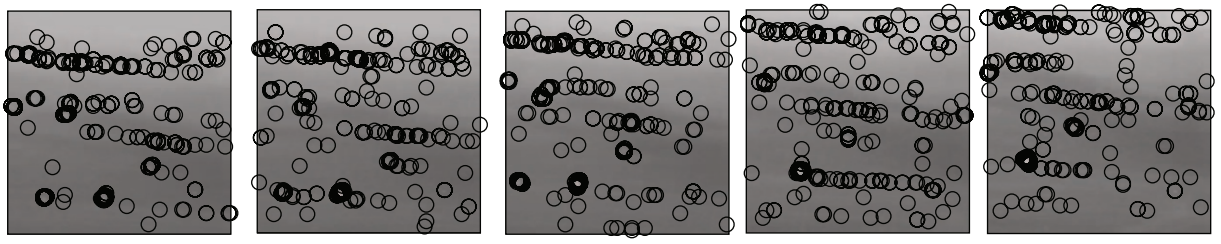
Frames 168 Frames 169 Frames 170 Frames 171 Frames 172

(a)



Likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$

(b)



Sampling

(c)

Figure 15: Common intermediate results for all the three tracking algorithms in a situation where the ego-motion compensation is especially challenging due to the aperture problem.

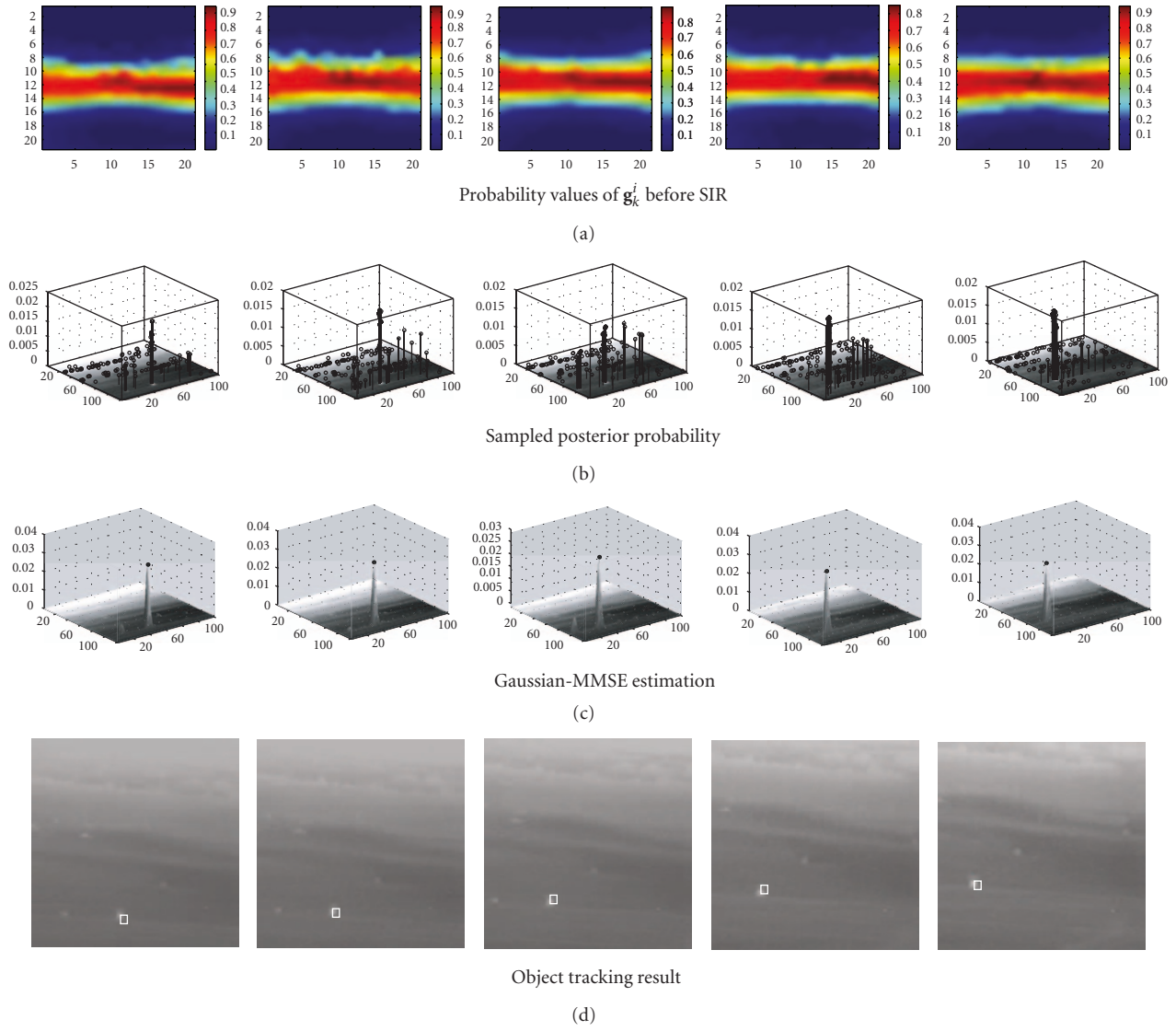


Figure 16: Tracking results for the BEH algorithm in a situation where the ego-motion compensation is especially challenging due to the aperture problem.

sequences acquired from a camera mounted on an airborne platform. A variety of moving and stationary terrestrial targets can be found in two different wavelengths: mid-wave (3 μm–5 μm) and long-wave (8 μm–12 μm). In general, the tracking task is quite challenging in this dataset due to the strong camera ego motion, the magnification and pose variations of the target signatures, and the own characteristics of the FLIR imagery described in Section 2.

In addition, the proposed object tracking algorithm has been compared with other two tracking algorithms to prove its superior performance using the same AMCOM dataset. These both algorithms are inspired on the existing works [8, 9], which also use a Particle Filter framework for the tracking, making easier and fairer to compare the performance of all the three algorithms. The three algorithms differ in the way they tackle the ego-motion: Bayesian modeling, deterministic modeling and not explicit modeling. The algorithm presented in this paper uses a Bayesian model for the ego-motion, and it is called tracking with Bayesian ego-motion handling (BEH). The second algorithm is based on a deterministic modeling and is referred to as tracking with deterministic ego-motion handling (DEH). It models the ego-motion by only one affine transformation, which is equivalent to express $p(\mathbf{g}_k)$ by a Kronecker's delta centered in $\mathbf{g}_k^d$, an affine transformation deterministically computed through the image registration algorithm described in [32]. The last algorithm, referred to as tracking with no ego-motion handling (NEH), has not an explicit model for the camera ego-motion, which leads to a simplified expression of the state transition probability:

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) = N(\mathbf{d}_k; \mathbf{M} \cdot \mathbf{d}_{k-1}, \sigma_{\text{tr}}^2), \tag{17}$$

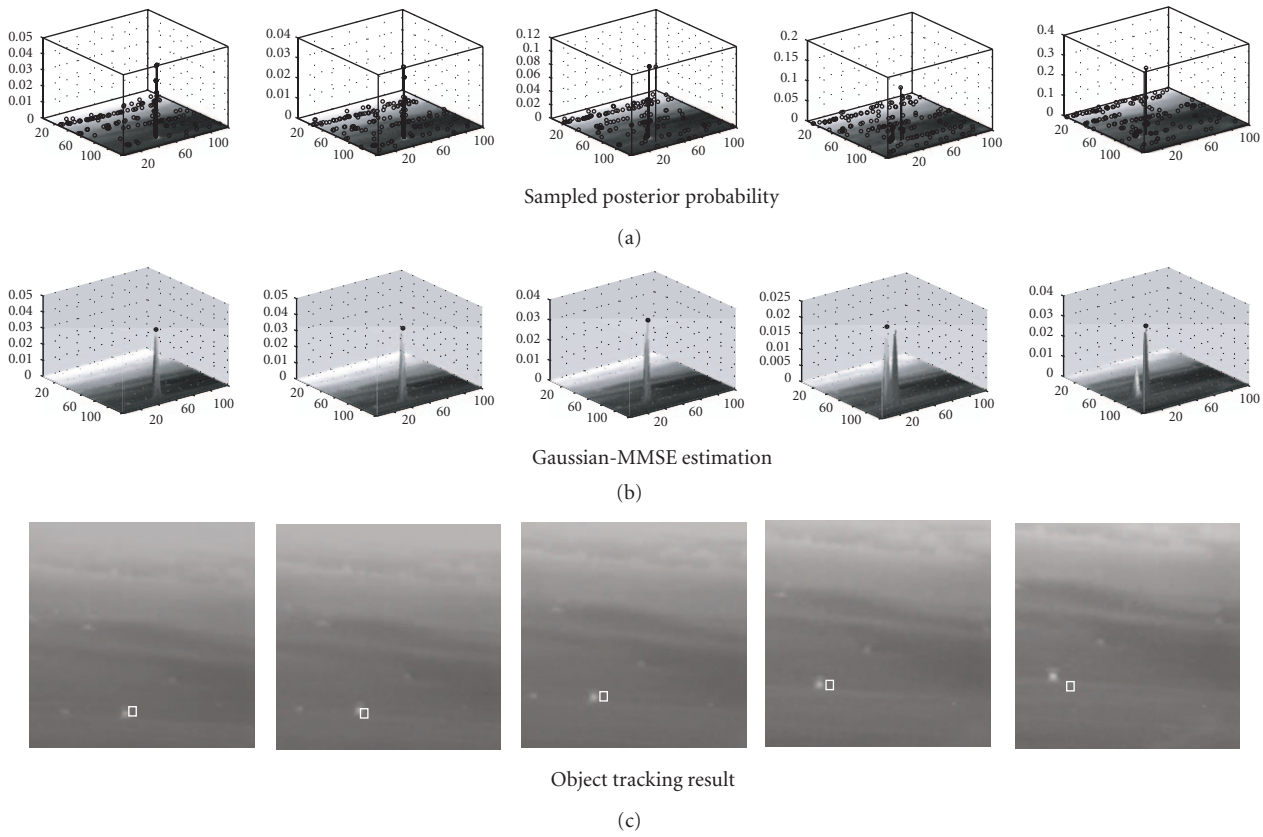


Figure 17: Tracking results for the DEH algorithm in a situation where the ego-motion compensation is especially challenging due to the aperture problem.

where the value of the parameter σ_{tr}^2 should be larger than that of BEH and DEH algorithms to try to alleviate the ego-motion effect.

With the purpose of making a fair comparison, the same number of samples has been used for the three algorithms: $N_S = 300$. This number is enough to ensure a satisfactory approximation of the state posterior probability given the specific characteristics of the AMCOM dataset. In the same way, the same value has been chosen for $\sigma_{\text{tr}}^2 = 2$ for the BEH and DEH algorithms, while a value of $\sigma_{\text{tr}}^2 = 4$ has been chosen for NEH algorithm, in order to alleviate its lack of an explicit ego-motion model and make it comparable with the other algorithms. The BEH algorithm needs an extra parameter which has been heuristically set to $\sigma_g^2 = 0.03$, offering good results for the given AMCOM dataset. However, other values with a variation less than the 15 percent have also offered similar results.

In the two following subsections, two different tracking situations are evaluated to demonstrate the higher performance of the BEH algorithm in complex ego-motion situations. The last subsection presents the overall tracking results for each of three algorithms using the aforementioned AMCOM dataset.

5.1. Strong Ego-Motion Situation. The BEH algorithm has been compared with the DEH and NEH ones for a situation of strong ego-motion. Figure 11 shows the common intermediate results for all the three algorithms. Figures 11(a) and 11(b) show two consecutive frames that have undergone a large displacement, in which the target object has been enclosed by a black rectangle as visual aid. Figure 11(c) shows the multimodal likelihood function, and lastly, Figure 11(d) shows the resulting Metropolis-Hasting based sampling, where each sample has been marked by a black circle.

Figure 12 shows the tracking results for the BEH algorithm. The probability values of $\mathbf{g}_k^i$ (the estimated affine transformations) before the SIR resampling are shown in Figure 12(a), which have been arranged in a rectangular grid, in a similar way to Figure 5. The probability values are displayed using a color scale. Notice that there is a peak in the middle left side, indicating that the camera has undergone a strong right translation motion. Figure 12(b) shows the sampled posterior probability, where the samples $\mathbf{d}_k^i$ with higher weights are correctly located over the target object, thanks to the Bayesian treatment of the camera ego-motion. Figure 12(c) shows the result of applying the Gaussian kernel over the sampled posterior probability, which is used by the Gaussian-MMSE estimator to compute the final state estimation (marked as a black circle). Finally, Figure 12(d) shows the target object satisfactorily enclosed by white rectangle, whose coordinates are determined by the state estimation. Observe that the infrared image is projected over

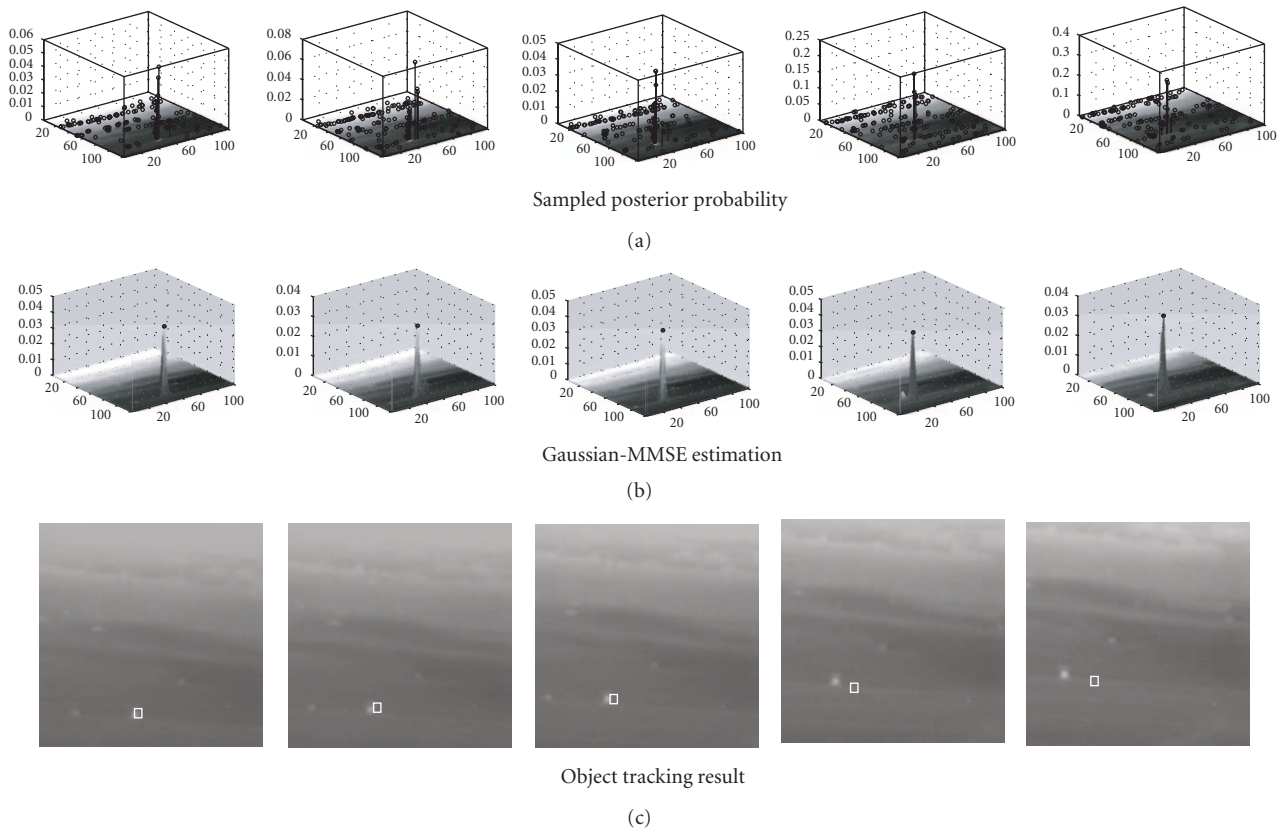


Figure 18: Tracking results for the NEH algorithm in a situation where the ego-motion compensation is especially challenging due to the aperture problem.

the X-Y plane of each probability distribution as visual aid.

Figures 13 and 14 show the tracking results for the DEH and NEH algorithms, respectively. Notice that the tracking fails in both cases, since the dynamic model does not correctly represent the camera and object dynamics, and consequently the tracking drifts to another mode of the likelihood function. In the case of DEH algorithm, this fact can be checked by observing that the estimated affine transformation corresponds to the one located in the coordinates $(11, 11)$ of Figure 12(a), which has a probability value much lower than the one related to the true camera motion.

5.2. High Uncertainty Ego-Motion Situation. A comparison or the tracking performance of all the three algorithms (BEH, DEH, and NEH) is presented for a situation where the ego-motion estimation is especially challenging due to the aperture problem (the frames are very low-textured). Figure 15 shows common intermediate results, in which the first column shows five consecutive frames, where the target object has been enclosed by a black rectangle as visual aid. The last two rows show the resulting multimodal likelihood function and the Metropolis-Hasting based sampling for each frame, respectively.

Figure 16 shows the tracking results for the BEH algorithm. The probability values of $\mathbf{g}_k^i$ (the estimated affine transformations) before the SIR resampling are shown in the first column, which have been arranged in a rectangular grid, in a similar way to Figure 5. The probability values are displayed using a color scale. Notice that there is not a well-defined peak, unlike the strong ego-motion situation (Figure 12(a)), but there is a set of affine transformation candidates with similar probability values, meaning that whatever of them could be the true camera motion. The affine transformations with higher probability value are located in the horizontal direction, indicating that the aperture problem is especially significant in that direction. In other words, the horizontal translation of the camera motion cannot be reliably computed between consecutive frames. The second column of Figure 16 shows the sampled posterior probability related to each frame. Notice that there are several samples with high weights that are not located over the target object, as a consequence of the high uncertainty in the camera ego-motion estimation. However, the majority of samples that have a high weight are located over the target object, allowing to track it satisfactorily. This fact can be verified by observing the two last columns, which, respectively, show the Gaussian-MMSE estimation and the tracking result, where the target object has been satisfactorily enclosed by a rectangle (whose coordinates are determined by the state estimation).

Figures 17 and 18 show the tracking results for the DEH and NEH algorithms, arranged in the same way of Figure 16.

Observe that the tracking fails in the frame 172 for the DEH algorithm, and also in the frame 171 for NEH algorithm. These failures arise from the accumulation of slight errors in the estimation of the object location, which, in turn, are caused by the poor characterization of the camera ego-motion.

5.3. Global Tracking Results. Finally, the global results about the performance of the BEH, DEH, and NEH algorithms using the sequences of the AMCOM dataset are shown in Table 1. The table is divided into two sections, showing the tracking results for long-wave and mid-wave infrared imagery, respectively. The first two columns show the sequence name and the target name, respectively. The third, fourth, and fifth columns show the first frame, the last frame, and the number of consecutive frames in which the target appears. The remaining columns show the performance of the BEH, DEH, and NEH algorithms, measured as the number of tracking failures and the tracking accuracy. The number of failures indicates the number of times that the target object has been lost. An object is considered to be lost when the rectangle that encloses the object according to the ground truth and the rectangle resulting from the tracking estimation do not overlap each other. The tracking accuracy has been defined as the average Euclidean distance between the object locations (centers of the corresponding rectangles) of the ground truth and the tracking estimation. Therefore, the accuracy will be better when its value is less. It is important to note that the algorithms are not reinitialized with ground truth data in case of tracking failure (object lost), since one of the more appealing advantages of the Particle Filter framework is its capability of recovering from tracking failures thanks to the handling of multiple hypotheses (or samples). This also affects the tracking accuracy, since all the erroneous object locations, derived from tracking failures, have been taken into account in its estimation. Therefore, the sequences with a lot of tracking failures will have a much worse tracking accuracy.

From the analysis of the number of tracking failures, it can be summarized that there are 11 situations (sequences) in which the BEH algorithm outperforms the DEH one and 16 situations in which the BEH algorithm outperforms the NEH one. Regarding the DEH algorithm, there are 11 situations in which it outperforms the NEH one, and 3 situations in which it outperforms BEH algorithm. Lastly, there are 4 situations in which the NEH algorithm outperforms the DEH one and none situation in which it outperforms the BEH algorithm. In the rest of situations, the performance is similar for all the three algorithms. To sum up, the BEH algorithm is the best of all, and the DEH algorithm is better than NEH one, as was expected. The errors obtained by the DEH and NEH algorithms arise from the poor characterization of the camera ego-motion, which is satisfactorily solved by the BEH algorithm.

The results about the tracking accuracy follow the same trend. An interesting fact happens when the ego-motion is quite low: the tracking accuracy of the DEH algorithm is slightly better than BEH one. The reason is that a deterministic approach introduces less uncertainty than a Bayesian approach for situations in which it is possible to reliably compute the camera motion. Nonetheless, the improvement is insignificant, and in addition, the BEH algorithm is able to cope with a wider range of situations than the rest.

There is one situation in which none of the three algorithms can ensure a correct tracking. This situation arises when the likelihood distribution has false modes very close to the true one (corresponding to the tracked object), and the apparent motion of the tracked object is very low. Under these circumstances, the tracker can be locked on a false mode.

As regards the type of infrared imagery, long and mid wave, the tracking results do not show any appreciable difference between them. Theoretically, mid-wave infrared imagery is better to detect and track objects with hot spots, arising from working engines and exhaust pipes, since the target-background contrast is greater. However, if the terrestrial vehicles are not working, and therefore they are at room temperature, the long-wave infrared imagery is preferable, since the target-background contrast is much greater. Anyway, the AMCOM dataset is not oriented to examine these kinds of differences, since each sequence is only acquired in a specific wave range, and therefore, a thorough comparison is not possible. Regarding the tracking performance, the only condition is that there exists an appreciable contrast between the target and the background, since the proposed Bayesian framework is able to handle the clutter (background regions with similar infrared signature to the target object) by means of the coherence between each object region candidate and the object and camera dynamics.

In order to provide a better understanding of the results presented in Table 1, the following website http://www.gti.ssr.upm.es/paper/RobustTracking/ has been built, which contains the object tracking results along with the ground truth for all the sequences. In addition, all the intermediate results (likelihood probability, MCMC sampling, probability values of the affine transformations, posterior probability, and Gaussian-MMSE estimation) are also available, which are useful to comprehend the obtained tracking results.

6. Conclusions

A novel strategy for object tracking in aerial imagery is presented, which is able to deal with complex situations in which the ego-motion cannot be reliably estimated. The proposed algorithm uses a complex dynamic model that combines the object and camera dynamics to predict the possible object locations. A probabilistic formulation is used to represent the camera dynamics by a set of affine transformations, each one corresponding to a possible camera ego-motion. Using this robust model to encode the dynamic information, the tracking algorithm is able to distinguish the actual object location among multiples candidates, derived from the appearance model of the object. This approach has been proven to be very robust not only in situations with strong ego-motion but also in those situations in which the ego-motion cannot be accurately estimated due

Table 1: Comparison of the tracking performance of the three algorithms BEH, DEH, and NEH using the AMCOM dataset. See the text for a detailed description of the results.

Sequence name	Target	First frame	Last frame	No. of frames	BEH		DEH		NEH	
					No. of failures	Tracking accuracy	No. of failures	Tracking accuracy	No. of failures	Tracking accuracy
Long-wave infrared imagery										
L19NSS	M60	1	57	57	15	6.19	0	4.10	14	6.78
L19NSS	mantruck	86	100	15	0	7.33	2	12.40	2	12.30
L19NSS	mantruck	211	274	64	0	4.90	0	6.69	0	4.98
L1415S	Mantruck	1	280	280	8	14.14	10	14.84	25	16.32
L1607S	mantrk	225	409	185	0	3.49	0	3.56	0	4.78
L1808S	apc1	1	79	79	0	2.11	0	2.19	0	1.99
L1808S	M60	1	289	289	0	2.95	0	2.74	1	3.73
L1808S	mantrk	193	289	97	0	3.93	0	3.38	48	9.09
L1618S	apc1	1	290	290	0	13.73	0	18.90	0	16.57
L1618S	M60	1	100	100	0	3.51	0	1.87	0	2.35
L1701S	Bradley	1	370	370	0	6.54	0	10.89	0	8.21
L1701S	pickup(trk)	1	30	30	0	1.99	0	1.99	0	1.70
L1702S	Mantruck	113	179	67	0	3.27	8	6.18	0	4.56
L1702S	Mantruck	631	697	67	0	2.57	0	2.43	0	2.59
L1720S	target	1	34	34	0	2.91	0	2.60	4	4.99
L1720S	M60	43	777	735	15	6.59	0	5.80	0	4.39
L605S	apc1	1	86	86	0	1.41	0	1.52	0	1.48
L605S	M60	615	641	27	0	4.88	0	4.80	19	13.15
L605S	tank1	614	734	125	0	2.14	0	2.17	163	65.73
L1812S	M60	72	157	86	0	2.53	0	2.52	0	2.71
L1813S	apc1	1	167	167	0	2.68	0	2.64	0	2.84
L1817S-1	M60	1	193	193	0	4.46	0	3.41	0	4.18
L1817S-2	M60	1	189	189	0	4.93	0	4.43	0	4.75
L1818S	apc1	21	112	92	32	6.01	72	10.22	72	26.05
L1818S	M60	81	202	122	60	10.55	119	17.79	118	85.25
L1818S	tank1	151	364	214	119	35.89	213	54.29	213	55.47
L1906S	Mantruck	1	203	203	0	8.47	0	5.58	0	8.21
L1910S	apc1	56	129	74	0	10.70	0	11.47	0	11.15
L1911S	apc1	1	164	164	0	8.15	0	9.19	0	8.75
L1913S	apc1	1	264	264	25	6.79	262	29.93	262	27.69
L1913S	M60	182	264	61	0	10.42	0	9.19	98	15.70
L1918S	tank1	26	259	234	15	7.03	20	8.98	144	9.46
L2018S	tank1	1	447	447	0	3.77	0	4.52	4	4.32
L2104S	bradley	1	320	320	51	5.60	319	34.39	319	43.82
L2104S	tank1	69	759	691	235	16.88	629	56.99	673	48.33
L2208S	apc1	1	379	379	0	5.88	0	5.40	0	5.53
L2312S	apc1	1	367	367	0	2.11	0	1.70	0	2.10

Table 1: Continued.

Sequence name	Target	First frame	Last frame	No. of frames	BEH		DEH		NEH	
					No. of failures	Tracking accuracy	No. of failures	Tracking accuracy	No. of failures	Tracking accuracy
Mid-wave infrared imagery										
M1406S	Bradley	1	379	379	0	1.84	0	1.92	0	2.72
M1407S	Bradley	1	399	399	0	4.83	0	5.77	0	4.37
M1410S	tank1	1	497	497	0	7.45	0	5.25	0	3.91
M1413S	Mantruck	1	379	379	2	7.44	1	7.73	0	6.67
M1415S	Mantruck	1	10	10	0	3.14	2	6.68	0	3.09
M1415S	Mantruck	15	527	513	7	5.42	512	62.30	512	65.16

to the aperture problem, strong camera motion, and/or the presence of independent moving objects. In these cases, it clearly outperforms other tracking approaches based on a deterministic ego-motion compensation or even without explicit compensation. The experimental results, performed with the AMCOM dataset, support this conclusion.

Acknowledgments

This work has been partially supported by the Comunidad de Madrid under project S-0505/TIC-0223 (Pro-Multidis) and by the Ministerio de Ciencia e Innovacion of the Spanish Government under project TEC2007-67764 (SmartVision).

References

[1] U. Braga-Neto, M. Choudhary, and J. Goutsias, "Automatic target detection and tracking in forward-looking infrared image sequences using morphological connected operators," *Journal of Electronic Imaging*, vol. 13, no. 4, pp. 802–813, 2004.

[2] H. Xin and T. Shuo, "Target detection and tracking in forward-looking infrared image sequences using multiscale morphological filters," in *Proceedings of the 5th International Symposium on Image and Signal Processing and Analysis (ISPA '07)*, pp. 25–28, September 2007.

[3] C. Wei and S. Jiang, "Automatic target detection and tracking in FLIR image sequences using morphological connected operator," in *Proceedings of the 4th International Conference on Intelligent Information Hiding and Multiedia Signal Processing (IIH-MSP '08)*, pp. 414–417, August 2008.

[4] A. Bal and M. S. Alam, "Automatic target tracking in FLIR image sequences," in *Automatic Target Recognition XIV*, vol. 5426 of *Proceedings of SPIE*, pp. 30–36, April 2004.

[5] A. Bal and M. S. Alam, "Automatic target tracking in FLIR image sequences using intensity variation function and template modeling," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 5, pp. 1846–1852, 2005.

[6] W. Yang, J. Li, D. Shi, and S. Hu, "Mean shift based target tracking in FLIR imagery via adaptive prediction of initial searching points," in *Proceedings of the 2nd International Symposium on Intelligent Information Technology Application (IITA '08)*, pp. 852–855, December 2008.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[8] N. A. Mould, C. T. Nguyen, and J. P. Havlicek, "Infrared target tracking with AM-FM consistency checks," in *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 5–8, March 2008.

[9] V. Venkataraman, G. Fan, and X. Fan, "Target tracking with online feature selection in FLIR imagery," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2nd edition, 2004.

[11] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[12] R. Kumar, H. Sawhney, S. Samarasekera, et al., "Aerial video surveillance and exploitation," *Proceedings of the IEEE*, vol. 89, no. 10, pp. 1518–1538, 2001.

[13] I. Cohen and G. Medioni, "Detecting and tracking moving objects in video from an airborne observer," in *Proceedings of the Workshop in Image Understanding*, pp. 217–222, 1998.

[14] B. Jung and G. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in *Proceedings of the International Conference on Intelligent Autonomous Systems*, pp. 980–987, 2004.

[15] C. R. del Bianco, F. Jaureguizar, L. Salgado, and N. García, "Aerial moving target detection based on motion vector field analysis," in *Proceedings of the 9th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS '07)*, vol. 4678 of *Lecture Notes in Computer Science*, pp. 990–1001, August 2007.

[16] C. R. del Blanco, F. Jaureguizar, L. Salgado, and N. García, "Target detection through robust motion segmentation and tracking restrictions in aerial FLIR images," in *Proceedings of the 14th IEEE International Conference on Image Processing (ICIP '06)*, vol. 5, pp. 445–448, September 2006.

[17] C. R. del Blanco, F. Jaureguizar, L. Salgado, and N. García, "Automatic aerial target detection and tracking system in airborne FLIR images based on efficient target trajectory filtering," in *Automatic Target Recognition XVII*, vol. 6566 of *Proceedings of SPIE*, Orlando, Fla, USA, April 2007.

[18] R. Pless, T. Brodsky, and Y. Aloimonos, "Detecting independent motion: the statistics of temporal continuity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 768–773, 2000.

[19] M. Irani and P. Anandan, "Video indexing based on mosaic representations," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 905–921, 1998.

[20] A. Yilmaz, K. Shafique, N. Lobo, X. Li, T. Olson, and M. A. Shah, "Target-tracking in FLIR imagery using mean-shift and global motion compensation," in *Proceedings of the Workshop on Computer Vision Beyond the Visible Spectrum*, pp. 54–58, 2001.

[21] A. Yilmaz, K. Shafique, and M. Shah, "Target tracking in airborne forward looking infrared imagery," *Image and Vision Computing*, vol. 21, no. 7, pp. 623–635, 2003.

[22] A. Strehl and J. K. Aggarwal, "Detecting moving objects in airborne forward looking infra-red sequences," in *Proceedings of the Workshop on Computer Vision Beyond the Visible Spectrum*, pp. 3–12, 1999.

[23] A. Strehl and J. K. Aggarwal, "MODEEP: a motion-based object detection and pose estimation method for airborne FLIR sequences," *Machine Vision and Applications*, vol. 11, no. 6, pp. 267–276, 2000.

[24] S. Lankton and A. Tannenbaum, "Improved tracking by decoupling camera and target motion," in *Real-Time Image Processing*, vol. 6811 of *Proceedings of SPIE*, San Jose, Calif, USA, January 2008.

[25] H. Zhang and F. Yuan, "Vehicle tracking based on image alignment in aerial videos," in *Proceedings of the 6th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR '07)*, vol. 4679 of *Lecture Notes in Computer Science*, pp. 295–302, August 2007.

[26] S. Ali and M. Shah, "COCOA—tracking in aerial imagery," in *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications III*, vol. 6209 of *Proceedings of SPIE*, Kissimmee, Fla, USA, April 2006.

[27] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, Berlin, Germany, 2006.

[29] 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.

[30] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[31] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, vol. 1, pp. 134–144, June 2003.

[32] S. Periaswamy and H. Farid, "Medical image registration with partial data," *Medical Image Analysis*, vol. 10, no. 3, pp. 452–464, 2006.

[33] J. Domke and Y. Aloimonos, "A probabilistic notion of correspondence and the epipolar constraint," in *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT '06)*, pp. 41–48, June 2006.

[34] J. Domke and Y. Aloimonos, "A probabilistic framework for correspondence and egomotion," in *Proceedings of the 2nd International Workshop on Dynamical Vision (WDV '06)*, vol. 4358 of *Lecture Notes in Computer Science*, pp. 232–242, May 2007.

[35] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.