Real-time tracking of multiple objects using adaptive correlation filters with complex constraints

Victor H. Diaz-Ramírez a,*, Viridiana Contreras a, Vitaly Kober b, Kenia Picos a

a Instituto Politécnico Nacional – CITEDI, Ave. del Parque 1310, Mesa de Otay, Tijuana, BC 22510, Mexico
b Department of Computer Science, Division of Applied Physics, CICESE, Carretera Ensenada-Tijuana 3918, Zona Playitas, Ensenada, BC 22860, Mexico

A R T I C L E   I N F O

Article history:
Received 8 May 2013
Received in revised form 9 July 2013
Accepted 18 July 2013
Available online 29 July 2013

Keywords:
Object recognition
Object tracking
Composite correlation filters
Adaptive filters
Graphics processing unit
Opto-digital correlators

A B S T R A C T

A real-time system for classification and tracking of multiple moving objects is proposed. The system employs a bank of composite correlation filters with complex constraints implemented in parallel on a graphics processing unit. When a scene frame is captured, the system splits the frame into several fragments on the base of a modeling kinematic prediction of target’s locations. The fragments are processed with a bank of adaptive filters. The filters are synthesized with the help of an iterative algorithm, which optimizes discrimination capability for each target. Using complex constraints in the filter design, multiple objects in the input frame can be detected and classified by analyzing the intensity and phase distributions on the output complex correlation plane for each fragment. The performance of the proposed system in terms of tracking accuracy, classification efficiency and time expenses is tested and discussed with synthetic and real input-scene sequences. The results are compared with those of common techniques based on correlation filtering.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Research in pattern recognition has recently received a notable attention due to a growing interest in developing imaging systems for assisting people in real life. Automatic object identification, video surveillance, vehicle navigation and object tracking are examples of pattern recognition applications [1–3]. Pattern recognition consists of effective detection, localization and classification of objects of interest in an input observed scene. These problems can be addressed by different approaches, for instance, by feature-based methods or by correlation filtering. In feature-based methods [3,4], the observed scene is firstly preprocessed to extract relevant features of potential targets. Next, the features are analyzed in order to make a decision. A main drawback of feature-based methods is that its overall performance depends on some ad-hoc decisions, which often require optimization [5].

On the other hand, correlation filtering is an attractive alternative to feature-based methods, and it is a suitable option for real-time applications [6,7]. Correlation filters possess a good mathematical basis, and they can be implemented by exploiting massive parallelism either in hybrid opto-digital correlators [6,8,9] or high speed in digital hardware such as Graphic Processing Units (GPUs) [7,10] or Field Programmable Gate Arrays (FPGAs) [11]. A correlation filter is a linear system. The coordinates of the system output maximum are estimates of the target coordinates in the observed scene [12,13]. An important feature of correlation filters is that they are able to recognize objects in highly cluttered and noisy environments; for instance, when input images are corrupted by additive and disjoint noise or when targets suffer from geometrical distortions such as in-plane rotations or scaling [14–19]. The filters can be designed by optimizing various performance criteria [20,21] and taking account models of signals and noise. For distortion invariant pattern recognition, composite filters are commonly used [22–24]. These filters are synthesized by combining several training templates, which represent a set of possible target views. Thus, a single composite filter can be used to recognize different target versions.

When a target moves across the observed scene, the target’s appearance with respect to the observer varies with time. Actually, target tracking consists in the estimation of target trajectory in the observed scene while the object moves across the environment [25,26]. Hence, the problem of target tracking can be treated with composite correlation filters applied to multiple frames. The tracking problem can be solved by detecting the object in the frames and by finding the correspondence between object states across the frames. Commonly object detection is carried out by employing feature-based methods [4,27], while tracking is performed by matching the states of detected objects in consecutive frames by taking into account a state-space model [28]. There are
several fruitful proposals which perform object tracking with the help of feature-based methods and state-space models [25,26,29,30]. However, these techniques face some unsolved issues when there are abrupt changes in target motion; target exits and reenters to the observed scene; the scene is distorted by additive and disjoint noise; the target is geometrically deteriorated by rotation and scaling. In order to overcome these difficulties, it is necessary to explore new approaches for target tracking. The use of correlation filtering for target tracking has been investigated as well.

Bhuiyan et al. [31,32] proposed a two-step approach for object classification and tracking. The first step in the approach is to use a bank of maximum average correlation height (MACH) filters [33] to detect potential targets within the scene. This is done by looking for correlation peaks in the outputs of the filter bank and then by taking the peak coordinates as location estimates for potential targets. Next, around the location estimates fragments of interest are formed from the current frame. It is assumed that each fragment contains one potential target of unknown class. In a second step, the fragments are processed with a bank of polynomial distance classifier (PDCC) filters [34] to establish the class for each detected object. This process is repeated for all frames. A main drawback of this proposal is that each frame is considered for each detected object. This process is repeated for all frames. A main drawback of this proposal is that each frame is considered for each detected object. This process is repeated for all frames. However, this proposal takes advantage of inherent parallelism of optics for processing large volumes of data in real-time. The correlator uses the number of elements imposed to the output cross-correlation value in the origin, d.

Recently, Manzur et al. [6] introduced a new opto-digital correlator for target detection, classification and tracking. This proposal takes advantage of inherent parallelism of optics for processing large volumes of data in real-time. The correlator employs a bank of binary phase-only (BPO) filters [35,36]. Each filter is able to recognize only one version of a target with low tolerance to geometrical distortions. The filters enter consequently and rapidly in the filter plane, and the system yields a sequence of output correlation responses. The obtained correlation planes are analyzed and processed to classify and to track the object across the frames.

In this work, we propose a real-time system for multiclass object recognition and tracking. The proposed system employs a bank of adaptive composite correlation filters with complex constraints. The filters are implemented in parallel on a GPU. First detection and classification of multiple targets are carried out by analyzing the correlation peaks at the system output for the current frame. Next, the system predicts states of the targets for the subsequent frame, and based on the prediction creates fragments of interest in the input frame and modifies the number of filters in the bank. When a new frame enters in the system, several small fragments around predicted locations for the targets are formed. The number of filters in the bank is also adjusted accordingly to predicted orientation of the targets in the current frame. Both location and orientation predictions are calculated by analyzing current and past state estimates and by taking into account a two-dimensional state-space motion model. The resultant system is able to perform classification and tracking of multiple moving objects in real-time by exploiting the benefits of massive parallelism of the GPU and by reducing false alarm probabilities by focusing the processing only on small fragments.

The paper is organized as follows. Section 2 presents a brief review of constrained composite filters for multiclass object recognition problems. In Section 3, we explain the proposed real-time system for multiclass object detection, classification and tracking. Section 4 presents the results obtained with the proposed system by testing its performance in synthetic and real-life input scene sequences. The obtained results in terms recognition performance, classification efficiency, tracking accuracy and speed of processing are presented. The results are compared with those of common techniques based on correlation filtering. Finally, Section 5 summarizes our conclusions.

2. Constrained composite filters for multiclass pattern recognition

Composite correlation filters use a set of training templates to represent multiple versions of a target in the observed scene. These filters are designed to produce a specific response in the presence of one of the known target versions used for training. Synthetic discriminant functions (SDFs) [22,27] yield a filter that is a linear combination of the training templates. Therefore, a SDF filter can be used for one-, two- and multiclass pattern recognition problems.

2.1. One-class problem

Let \( T = \{ g_i(x, y) \}_{i=1, \ldots, N_T} \) be a set of \( N_T \) different templates, each of them represents a different version of a given target \( s(x, y) \). A SDF filter is able to recognize different versions of the target with only one correlation operation. Let \( h(x, y) \) be the impulse response of a SDF filter given by

\[
h(x, y) = \sum_{i=1}^{N_T} a_i g_i(x, y),
\]

where \( a_i \) are unknown weighting coefficients. The filter \( h(x, y) \) is a linear combination of \( N_T \) training templates \( g_i(x, y). \) The weighting coefficients are calculated subject to the inner-product constraints

\[
(h^*(x, y), g_i(x, y)) = c_i,
\]

where \( * \) means complex conjugate and \( c_i \) is a prespecified quantity imposed to the output cross-correlation value in the origin, between filter \( h(x, y) \) and template \( g_i(x, y). \) Let \( Y \) be a \( N_T \times d \) matrix, where \( d \) is the number of elements in each template. The \( i \)th column in \( Y \) is given by \( g_i, \) a \( d \times 1 \) vector obtained by reordering elements of \( g_i(x, y) \) in lexicographical order. Let \( a \) and \( c \) represent column vectors of \( a_i \) and \( c_i \). Using matrix-vector notation, the filter \( h(x, y) \) and constraints \( c_i \) can be rewritten, respectively, as

\[
h = Ya,
\]

\[
c^* = Y^+ c.
\]

where superscript \( ^+ \) means conjugate transpose. By combining Eqs. (3) and (4), the solution vector (if matrix \( (Y^+ Y)^{-1} \) is non-singular) is

\[
h = Y(Y^+ Y)^{-1} c^*.
\]

The SDF filter in Eq. (5) can be used for one-class pattern recognition problems, that is, to recognize different versions of a given target \( s(x, y) \). This can be done by setting all elements of vector \( c \) equal to unity, i.e., \( c = [1, 1, \ldots, 1]^T \).

2.2. Two-class problem

Assume that we have a true-class set \( T \) and a false-class set \( F = \{ p_i(x, y) \}_{i=1, \ldots, N_F} \) consisting of \( N_F \) templates \( p_i(x, y) \) (each one with \( d \) elements). The latter represents unwanted patterns to be rejected. Let \( p \) be a \( d \times 1 \) vector containing the elements of \( p_i(x, y) \) in lexicographical order.

We can synthesize a two-class SDF filter which is able to recognize all target versions in \( T \) and to reject unwanted patterns in \( F \), by solving Eq. (5). Here matrix \( Y \) is a \( d \times (N_T + N_F) \) matrix,
given by
\[ \mathbf{Y} = [\mathbf{g}_{1}, \mathbf{g}_{2}, \ldots, \mathbf{g}_{N_{T}}, \mathbf{p}_{1}, \ldots, \mathbf{p}_{N_{F}}]^{T}, \]
and the vector \( \mathbf{c} \) is given by
\[ \mathbf{c} = \begin{bmatrix} 1, 1, \ldots, 1, 0, \ldots, 0 \end{bmatrix}^{T} \]

2.3. Multiclass problem

Suppose that we have \( Q \) different target classes to be recognized and classified by a filter \( h(x, y) \). The \( q \)th target class is represented by a set of templates \( \mathbf{G}^{(q)} = [\mathbf{g}^{(q)}_{i}(x, y) | i = 1, \ldots, N_{T}^{(q)}] \), where \( \mathbf{g}^{(q)}_{i}(x, y) \) is the \( i \)th training template that belongs to the \( q \)th target class. The true-class set \( T \) is now the union of \( Q \) subsets,
\[ T = \bigcup_{q=1}^{Q} \mathbf{G}^{(q)}. \]
The objective is to synthesize a SDF filter, which is able to recognize all target versions from \( Q \) classes in \( T \) and to reject all unwanted patterns in \( Q \). Additionally, the filter should identify the class of each recognized target. Let \( \mathbf{Y} \) be the \( d \times (N_{P} + \sum_{q=1}^{Q} N_{T}^{(q)}) \) matrix composed by
\[ \mathbf{Y} = [\mathbf{g}_{1}^{1}, \mathbf{g}_{1}^{2}, \ldots, \mathbf{g}_{1}^{Q}, \mathbf{g}_{2}^{1}, \ldots, \mathbf{g}_{2}^{Q}, \ldots, \mathbf{g}_{N_{T}}^{1}, \mathbf{g}_{N_{T}}^{2}, \ldots, \mathbf{g}_{N_{T}}^{Q}, \mathbf{p}_{1}, \ldots, \mathbf{p}_{N_{F}}]^{T}, \]

where \( \mathbf{g}_{i}^{q} \) is a \( d \times 1 \) vector representing the \( i \)th training template for the \( q \)th target class. Here, the \( q \)th target class contains \( N_{T}^{(q)} \) training templates and the false-class contains \( N_{P} \) templates. To recognize and to classify objects belonging to \( Q \) classes, a complex-valued vector \( \mathbf{c} \) is used with unitary magnitudes and specific phase values \( \psi_{j}^{q} |q = 1, \ldots, Q\), each of them assigned to one particular target class as follows [38]:
\[ \mathbf{c} = [\mathbf{c}_{1}^{1}, \mathbf{c}_{1}^{2}, \ldots, \mathbf{c}_{1}^{Q}, \mathbf{z}]^{T}, \]
and the vector \( \mathbf{c} \) is given by
\[ \mathbf{c} = \begin{bmatrix} \psi_{11}^{1}, \psi_{12}^{1}, \ldots, \psi_{1Q}^{1} \end{bmatrix}^{T}, \]
are prespecified vectors of constraints for each target class for \( |q = 1, \ldots, Q\), and \( \mathbf{z} \) is the \( N_{P} \times 1 \) vector of constraints (having only zeros) for the false class. The target constraints in Eq. (11) are complex quantities that uniquely identify the class of each detected object. Using procedure shown in Fig. 1 we can localize and classify targets from different classes. The first step is to calculate the complex correlation function \( c(x, y) \) between the input image \( f(x, y) \) and a different multiclass SDF filter \( h(x, y) \). Next correlation peaks in the intensity correlation plane \( |c(x, y)|^{2} \) are found. The coordinates of the peaks are taken as estimates for the location of the targets. To determine the class for each detected object, we evaluate the phase value in the complex function \( c(x, y) \) only at the coordinates of the estimated target positions. If the resultant phase value \( \phi \) is within the interval \( \{\psi_{q}^{q}, \psi_{q}^{q}+\epsilon_{q}\} \) then the target is assigned to the \( q \)th target class previously specified in the filter constraints.

3. Real-time system for object classification and tracking

In this section we describe a real-time system for object detection, classification and tracking. Let us consider an optical setup given in Fig. 2. Assume that a target moves in horizontal direction of the two-dimensional plane. At time \( t_{k} \), the camera captures a scene-frame containing the target with a orientation angle \( \theta_{k} \). The target is embedded into a background at the coordinates \( (x_{k}, y_{k}) \). Next at time \( t_{k+1} \) (when camera is ready to capture consequent frame) the target has moved to a new position with coordinates \( (x_{k+1}, y_{k+1}) \) and the orientation angle has changed from \( \theta_{k} \) to \( \theta_{k+1} \). We want to design a tracking system to estimate in real-time the sequence of target positions and orientation angles (sequence of states) as a function of time \( \{t_{k} = k\Delta_{t} | k = 1, 2, \ldots, \} \), where \( \Delta_{t} \) is the sampling interval. The proposed tracking system contains the following stages: observation, estimation, prediction and adaptation, as it is shown in the block diagram in Fig. 3.

**Observation stage**

The goal of this stage is to extract useful information from the observed scene that helps us to detect reliably targets as well as to estimate accurately the targets states. Let \( f_{k}(x, y) \) be an observed scene-frame captured at time \( t_{k} \). The observed frame can be described under the nonoverlapping signal model as follows [14]:
\[ f_{k}(x, y) = \sum_{q=1}^{Q} \mathcal{C}_{k}^{q}(x - r_{k}^{q}, y - r_{k}^{q}; \theta_{k}^{q}) + b(x, y) \sum_{q=1}^{Q} \mathcal{W}_{k}^{q}(x - r_{k}^{q}, y - r_{k}^{q}; \theta_{k}^{q}) + n_{k}(x, y), \]
where \( x \in [1, N] \) and \( y \in [1, M] \) are the spatial coordinates, and \( N \) and \( M \) are the limits of observable area. According to Eq. (12), the \( k \)th frame can be formed by superposition of objects belonging to \( Q \) classes \( \{s_{k}(x, y) | q = 1, \ldots, Q\} \), which are embedded into a disjoint background \( b(x, y) \) at unknown coordinates \( (r_{k}^{q}, \theta_{k}^{q}) \). Each frame may contain the \( q \)th target rotated by the orientation angle \( \theta_{k}^{q} \). Furthermore, \( \mathcal{W}_{k}^{q}(x, y) \) is a binary function of the inverse region of support of \( s_{k}(x, y) \), and \( h(x, y) \) is a realization of a stationary random process representing additive sensor’s noise. The desired tracking system, should estimate in real-time the correct state of each target by processing the observed scene-frames \( f_{k}(x, y) \). The state of the \( q \)th target in time \( t_{k} \) is represented by a state vector \( \mathbf{s}_{k}^{q} = [x_{k}^{q}, y_{k}^{q}, \theta_{k}^{q}]^{T} \), which contains the target position \( (x_{k}^{q}, y_{k}^{q}) \) and the orientation angle \( \theta_{k}^{q} \).

**Fig. 1.** Block diagram for location and classification of targets using a multiclass composite filter with complex constraints.
patterns in Eq. (12) when observation stage. First, let us consider the simplest case in one class with only one viewing angle. Since the target and the background are nonoverlapped, Eq. (12) can be rewritten as
\[ Q(\bar{x} | \bar{g}) = \arg \max \left\{ \frac{1}{2} \sigma^2 \left( f_T^T(x) - S(x - \bar{x}) \right)^2 \right\}, \]
where \( C \) is a normalization constant. In this manner, by substitution of Eq. (19) into Eq. (18) and after some manipulations, we get
\[ \hat{\bar{x}} = \arg \max_x \left\{ \sum_x f_T^T(x) S(x - \bar{x}) + 2\sigma^2 \ln[P(x)] \right\} = \arg \max_x [f_T^T(x) \otimes S(x) + 2\sigma^2 \ln[P(x)]] \]
where the symbol “\( \otimes \)” represents the linear correlation. Observe that Eq. (20) is the MAP estimator for the target location in \( f_T^T(x) \). The estimate gives the position of the maximum value in the cross-correlation function between the reference template \( s_T(x) \) and the observed signal \( f_T(x) \), biased with the logarithm of the a-priori probability distribution of the target location. If a-priori information about the target location is unavailable then it may be assumed that the distribution of the target location in the input scene is uniform. In this case the MAP estimator of the target location depends only on the correlation term in Eq. (20). The impulse response of the linear system is given by the Matched Spatial Filter (MSF) [14,39]. The MSF filter is a correlation filter that is optimum with respect to the signal to noise ratio (SNR) when a target is corrupted with additive noise. Also, it has been shown [13] that the MSF yields unbiased estimates for the target location and yields the minimum variance in terms of localization errors (LEs). However, the MSF has a poor tolerance to disjoint noise and to geometrical degradations of targets considered in Eq. (12). To overcome this limitation one of two common approaches can be adopted. The first approach is given by optimization of a correlation filter in terms of performance metrics. The second approach uses a post-processing step before taking a decision [40]. For high-speed applications such as target tracking the former approach is preferable because it does not require additional computing time for post-processing. In this work, we use an adaptive approach [41] for optimization of the correlation system based on the MAP estimator. Basically, the adaptive approach for the filter design consists in iterative training of a basic composite filter in order to obtain a correlation filter with a good tolerance to geometric distortions of targets and with a good robustness to additive and disjoint noise. Thus we can design a composite filter as a combination of several MSFs.

Suppose that \( f_T^T(x,y) \) in Eq. (14) has a single target whose location and orientation are functions of time. Assume that \( f_T^T(x,y) \) is a function representing disjoint background noise. The estimation of the target state in \( f_T^T(x,y) \) can be performed by solving a two-class pattern recognition problem (see Section 2.2), where all target versions in \( f_T^T(x,y) \) form the true-class, and the background is the false-class. Furthermore, when \( f_T^T(x,y) \) contains several target versions belonging to different classes, the problem is a multiclass pattern recognition task (see Section 2.3). Solutions can be obtained with the help of MAP estimator in Eq. (20) by replacing the basic MSF filter with an adaptive multiclass SDF filter trained with appropriate representations of the true- and false-class templates in sets \( T \) and \( F \). Note that the training templates for the true-class objects can be easily chosen a-priori.
False-class templates can be obtained with the help of an adaptation algorithm [41,42].

Assume that the true-class templates \( f_{g_q^i}(x,y) \in G^q \) are given, and the false-class templates \( f_{p_i}(x,y) \in F \) are unknown. Any template \( p_i(x,y) \) can be substituted by an object having similar structures to those of the false-object. If the background to be rejected is available the false-class templates \( p_i(x,y) \) can be obtained by cutting out fragments from the background. We use an adaptive iterative algorithm to find \( F \). The first step of the algorithm is to calculate the cross-correlation between the background image and a basic multiclass SDF filter (Section 2.3), initially trained with all available templates of the target and all known false-class objects. The background function can be either described as a picture or by a stochastic process. Next, we search for the coordinates of a maximum sidelobe in the output correlation plane. The goal is to incorporate a small fragment from the background image (around the location of the maximum sidelobe) as a new false-class template \( \hat{p}(x,y) \) in \( F \).

A new false object \( \hat{p}(x,y) \) should improve the performance of the composite filter in terms of the discrimination capability (DC) [12]. The DC is defined as the ability of a filter to distinguish among true- and false-class objects; it is formally defined by

\[
DC = 1 - \frac{|c_{F_{\text{max}}}^e|^2}{|c_{T_{\text{max}}}^e|^2},
\]

where \( |c_{F_{\text{max}}}^e|^2 \) is the maximum intensity value in the output correlation plane over the background area, and \( |c_{T_{\text{max}}}^e|^2 \) is the maximum intensity value in the output correlation plane over the area occupied by the target. Note that the two areas are complementary. A filter with a DC close to unity has a good capacity to distinguish objects from true and false classes. Negative values of the DC mean that the filter is unable to recognize targets. Next, after including a new template \( \hat{p}(x,y) \) in \( F \) a new multi-class SDF filter is synthesized. This procedure is iteratively repeated until a prespecified value of the DC is reached. A block diagram of the adaptation procedure is shown in Fig. 4, and steps are summarized below.

- **Step 1:** Assign all available training templates to a corresponding target class: \( \{g_q^i(x,y) \mid i = 1 \ldots N_q \} \rightarrow G^q \) and construct matrix \( \Upsilon \) (see Eq. (9)). Assign \( \psi^q \) value to each object of the \( Q \) different target-classes: \( \psi^q \rightarrow G^q \) for \( q = 1 \ldots Q \), and construct vector \( c \) using Eqs. (10) and (11).

- **Step 2:** Synthesize a multiclass SDF filter trained for sets \( T \) and \( F \) with the help of Eq. (5).

- **Step 3:** Calculate cross-correlation between the multiclass SDF filter and the background image.

- **Step 4:** Calculate the DC value for each template of set \( G^q \). Then, set the minimum DC value as the current DC value of the filter. If the DC value of the filter is greater than a prespecified value, the procedure is finished. Otherwise, go to next step.

**Fig. 4.** Iterative algorithm used for synthesis of adaptive multicase SDF filters.

**Fig. 5.** Block diagram of the proposed system for detection, classification and tracking of moving objects.
We expect that a multiclass SDF filter obtained with the above procedure, can be used to detect and to estimate the location of several targets belonging to different classes using the procedure shown in Fig. 1. It is important to realize that if there is a broad range of geometrical distortions for the targets, a bank of filters also will be large in order to achieve a high performance in terms of the DC.

Estimation of target view angle

Let \( \{ h_i(x, y) \}_{i = 1, \ldots, N_h} \) be a filter bank containing \( N_h \) multi-class SDF filters trained with the adaptive procedure. Each filter is able to detect targets from \( Q \) different classes within a small range of in-plane rotations specified by \( \{ \theta_i \pm \epsilon_i \} \). Note that for \( Q \) classes, always \( Q \leq Q \). Suppose that we deal with a pattern recognition problem with \( Q = 2 \), and the targets from each class can be rotated freely within the range of \( 0^\circ \) to \( 360^\circ \). By setting \( Q = 2 \), \( \epsilon_i = 5^\circ \), and \( \theta_i = \{ 0, 10, 20, \ldots, 350 \}^\circ \), we obtain a filter bank with \( N_h = 36 \) composite filters. Each filter is able to recognize targets from \( Q = 2 \) classes with a specific rotation angle of \( \theta_i \) and with tolerance of \( \pm 5^\circ \). For instance, if a target in the input scene is rotated by \( 3^\circ \), it is expected that only filter \( h_1(x, y) \) will correctly detect the target. So, with this approach we can estimate a small sector (or interval) which contains the rotation angle of a target in the current frame. This sector can be estimated as \( 2\pi r/360^\circ \pm \epsilon_i \) degrees, where \( r \) is the index of the filter in the bank that detected the target in the current frame with the highest DC value. Several algorithms were suggested to precisely estimate the angle of rotation of a target with filter banks \( [43-45] \). Figue and Refregier \( [44] \) proposed a method that creates a response which is approximately linear with respect to the target angle in a small range of rotations. This method can be utilized to precisely estimate the angle \( \theta \).

Prediction stage

The output of the estimation stage for each target in \( f_k(x, y) \) is an estimated state vector \( \vec{f}_k = [x_k, y_k, \theta_k] \). In order to improve the state estimation for the subsequent frame \( f_{k+1}(x, y) \) we take into account available information from the current and past estimates to predict subsequent vectors \( \vec{f}_{k+1} = [x_{k+1}, y_{k+1}, \theta_{k+1}] \). Note that this prediction will help to make faster and more accurate state estimations and to match the targets states across observed frames. To predict the states in successive frames we can characterize motion behavior of targets with an appropriate state-space motion model. According to the optical setup shown in Fig. 2, targets move on two-dimensional plane. Therefore, the motion behavior of the \( q \)-th target can be characterized by a coordinated turn model \( [25] \), which obeys the following state-space equations:

\[
\begin{align*}
\vec{x}_{k+1} &= \vec{x}_k + \frac{\sin(\omega_{k}^\Delta \Delta_k)}{a_k} x_k - \frac{1 - \cos(\omega_{k}^\Delta \Delta_k)}{a_k} y_k + a_k^{\Delta^2/2} \\
\vec{y}_{k+1} &= \vec{y}_k + \frac{1 - \cos(\omega_{k}^\Delta \Delta_k)}{a_k} x_k + \frac{\sin(\omega_{k}^\Delta \Delta_k)}{a_k} y_k + a_k^{\Delta^2/2} \\
\vec{\theta}_{k+1} &= \omega_{k}^\Delta \Delta_k + \theta_k + a_k^{\Delta^2/2} \\
\end{align*}
\]

where \( \omega_{k}^\Delta \Delta_k = \cos(\omega_{k}^\Delta \Delta_k) \theta_k - \sin(\omega_{k}^\Delta \Delta_k) \gamma_k + \delta^\Delta_k \theta_k \)

\( a_k^{\Delta^2/2} \) is the diffusion coefficient.
The variables $x_k^q$ and $y_k^q$ represent the position of the $q$th target in Cartesian coordinates, $\dot{x}_k^q$ and $\dot{y}_k^q$ are velocity components in $x$ and $y$ directions, and $\omega_k^q$ is the target's angular rate. Furthermore, $a_{x,k}^q$ and $a_{y,k}^q$ are random variables representing acceleration components in $x$ and $y$ directions and $\omega_{\omega,k}^q$ is the angular acceleration. In our motion model, suppose that acceleration components are caused by perturbations such as turbulence or inhomogeneous friction. The position of the target in a subsequent time, can be predicted by substitution of the estimated position $\hat{x}_k^q; \hat{y}_k^q$, the estimated velocity components $(\hat{x}_k^q; \hat{y}_k^q)$, and the estimated turn rate $\hat{\omega}_k$ (all calculated from current and past frames) into the state space equations in Eq. (22), and then by taking the expected value. Observe, that the velocity components can be easily approximated as $\dot{x}_k^q = \dot{x}_k^q / C_0$ and $\dot{y}_k^q = \dot{y}_k^q / C_0$. The angular rate $\dot{\omega}_k$ is calculated in each frame as the magnitude of the angular acceleration $\omega_{\omega,k}$ divided by the target speed $v_k$ [25].

**Adaptation stage**

The adaptation stage performs convenient adjustments for the size of the input frame and for the number of filters in the bank based on state predictions made in each frame. First, by using predicted positions for targets, the system cuts out from the input frame several fragments of interest, each of them contains at least one target of unknown class. Since the size of resultant fragments is much smaller than the size of the input frame, the speed of processing increases considerably because state estimation is carried out only on small fragments. Additionally, by the use of prediction of orientation states of each target, the system determines the required number of correlation filters to perform state estimations in consequent frame.

### 3.1. Proposed system for detection, classification and tracking of multiple objects

The block diagram of the proposed system is depicted in Fig. 5, and operation steps are explained (for one target) and summarized below.

- **Step 1:** Read a scene frame $f_k(x,y)$ with size $N_x \times N_y$ from the input observed sequence.
- **Step 2:** Set initial adaptation parameters as follows: the center of the fragment of interest is $x_0 = N_x / 2$ and $y_0 = N_y / 2$, the size of the fragment is $\bar{N}_x = N_x$ and $\bar{N}_y = N_y$, and the number of filters to be used in the bank is $\bar{K} = N_b$. Note that these parameters indicate that the state of the target will be estimated from the observed scene frame using all available correlation filters.

![Fig. 8. Example of scene frames used in experiments in environment of 10 dB additive noise SNR. (a) frame 1, (b) frame 4, (c) frame 7, (d) frame 10, (e) frame 13, (f) frame 16, (g) frame 174, (h) frame 175, (i) frame 176.](image-url)
Table 1

<table>
<thead>
<tr>
<th>Objects</th>
<th>DC</th>
<th>LE</th>
<th>(\hat{\phi})</th>
<th>(\hat{\theta})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>0.75</td>
<td>0</td>
<td>46.1(^{\circ})</td>
<td>270(^{\circ})</td>
</tr>
<tr>
<td>Target 2</td>
<td>0.75</td>
<td>0</td>
<td>90.3(^{\circ})</td>
<td>110(^{\circ})</td>
</tr>
</tbody>
</table>

**Step 3:** Cut off a fragment of interest \(f_s(x, y)\) from current frame \(f_s(x, y)\) accordingly with adaptation parameters.

**Step 4:** Compute the Fourier Transform (FT) of subframe \(f_s(x, y)\)

**Step 5:** Compute the Inverse Fourier Transform (IFT) for each filtered spectrum \(\mathcal{F}_f(\mu, \nu)\) and obtain the set of complex correlation planes \(\{c(x, y) = IFT[\mathcal{F}_f(\mu, \nu)\mathcal{H}_n(\mu, \nu)]\}\); for \(i = 1, ..., K\).

**Step 6:** Calculate the DC value for each resultant correlation intensity plane \(|c(x, y)|^2\) and find the plane with the highest DC value: \(c(x, y) = \max(|c(x, y)||DC| = DC_{\text{MAX}})\). Set the DC value of the chosen correlation plane as the current DC value.

**Step 7:** Find the coordinates of the estimated target location as \((\hat{x}_0, \hat{y}_0) = \arg \max_{x, y}(|c(x, y)|^2)\) and estimate the orientation angle of detected target \(\hat{\theta}\).

**Step 8:** Evaluate the phase value \(\hat{\phi}\) at the estimated target coordinates \((\hat{x}_0, \hat{y}_0)\) in the complex correlation plane \(c(x, y)\). Then, verify if \(\hat{\phi} \in [\phi^u, \phi^l]\). If the result is TRUE, assign the detected object to the target class and set the estimated state vector for current frame as \(\hat{\mathbf{s}} = [\hat{x}_0, \hat{y}_0, \hat{\phi}, \hat{\theta}]\); then go to next step. If the result is FALSE, the state of the target cannot be estimated; therefore, go to Step 1.

**Step 9:** Predict subsequent state vector \(\hat{\mathbf{s}}_{k+1} = [\hat{x}_p, \hat{y}_p, \hat{\phi}_p, \hat{\theta}_p]^T\) from \(\hat{\mathbf{s}}_k\) and update the adaptation parameters as follows: the center of the scene fragment is \(x_0 = x_p\) and \(y_0 = y_p\); the size of the scene fragment is \(N_x = N_x/A\) and \(N_y = N_y/B\) (for \(A\) and \(B\) constants), the number of filters to be used in the bank is \(K = K/C\) (for \(C\) constant), and the filter indices \(r(i) = 1, ..., K\) for the bank are chosen accordingly to predicted orientation state \(\hat{\theta}_p\).

**Step 10:** Read a new scene frame \(f_s(x, y)\) with size \(N_x \times N_y\) from the input observed sequence, then go to Step 3.

The proposed tracking system performs fast and accurate state estimations for moving targets by focusing signal processing only on small fragments of interest taken from the observed scene. Note if a state estimation cannot be correctly carried out using small fragments, then the estimation is performed using the original entire scene frame and all correlation filters in the bank, before observing consequent frame.

## 4. Results

In this section results obtained with the proposed system for detection, classification and tracking of multiple moving objects in noisy scenes are analyzed and discussed. The results are compared with those obtained with the MACH-PDCC approach proposed by Bhuiyan et al. [31,32] and with the system based on BPO filters proposed by Manzur et al. [6]. The performance of all systems is evaluated in terms of efficiency of target detection and classification, tracking accuracy, and speed of processing when the systems are tested in synthetic and real input-scene sequences. Detection performance is measured in terms of the DC, whereas classification performance is characterized by a minimum error rate in classifying detected objects. Tracking accuracy is characterized by the precision of estimates carried out for target states across scene frames. The accuracy in location estimation of targets is characterized by the LE, which is given by [13]

\[
LE = [(x^q - \hat{x}^q)^2 + (y^q - \hat{y}^q)^2]^{1/2}.
\]

Here, \((x^q, y^q)\) and \((\hat{x}^q, \hat{y}^q)\) are the exact (true) and estimated coordinates of location of the \(q\)th detected target, respectively. The accuracy of estimation of the orientation angle is characterized by the orientation error (OE), defined by

\[
OE = |\phi^q - \hat{\phi}^q|,
\]

where \(\phi^q\) and \(\hat{\phi}^q\) are the true and estimated orientation angles, respectively, for \(q\)th detected target. The speed of processing is measured in terms of the average frames-per-second (A-FPS) that tracking systems are able to process. In our experiments we use monochrome images with 640 × 480 pixels. The signal range is \([0, 1]\) with 256 quantization levels. The test patterns are shown in Fig. 6. They consist in two targets (car objects) and in a real-life...
background. In our tests, targets are free to move in the horizontal plane over the background at unknown coordinates, while its orientation angle is varying. Note that this scenario is consistent with the optical setup depicted in Fig. 2. The size of the targets is about 25 x 50 pixels with the mean value and standard deviation of $\mu_1 = 0.24$, $\sigma_1 = 0.26$ for target 1, and $\mu_2 = 0.14$, $\sigma_2 = 0.18$ for target 2. The background image has a mean value of $\mu_b = 0.3$ and a standard deviation of $\sigma_b = 0.1$. We construct a sequence of 400 scene frames using the test patterns shown in Fig. 6. Each frame is constructed accordingly with the nonoverlapping signal model described in Eq. (12) for different additive noise SNR values.

The trajectory of target states for 400 scene frames are obtained by means of the kinematics model given in Eq. (22) by varying parameters of velocity $v_k$ and angular rate $\omega_k$. The velocity parameters are varied within the range of 1-16 pixels-per-second across scene frames. Furthermore, acceleration parameters are given by random variables with a Gaussian distribution [25]. The resultant trajectories for two targets are shown in Fig. 7. Note that a location state in Fig. 7 is represented by the coordinates of a circle for target 1 and by the coordinates of a diamond for target 2. The orientation state is indicated by the angle formed between the line connected to a location point and vertical axis. Fig. 8 shows examples of various scene frames when corrupted with 10 dB SNR additive noise. It is important to note that created sequence contains situations when targets suddenly exit the scene and reenter from another point (see frames 174–176). First, let us illustrate the detection and classification performance of adaptive multiclass SDF (AMSDF) filters.

To this purpose, we design a filter bank trained to recognize and classify two targets with tolerance to in-plane rotations inside the range of [0, 360]$^\circ$. 36 AMSDF filters with complex constraints were synthesized with the help of the iterative algorithm shown in Fig. 4. Each AMSDF filter is able to recognize and classify two targets with a specific orientation angle $\theta$ with a tolerance of $\pm 5^\circ$. In this manner, filter 1 is able to recognize and classify two targets rotated by $0^\circ \pm 5^\circ$, filter 2 can recognize and classify two targets rotated by $10^\circ \pm 5^\circ$, filter 19 is able to recognize and classify two targets rotated by $180^\circ \pm 5^\circ$, and filter 36 can recognize and classify two targets rotated by $350^\circ \pm 5^\circ$. All adaptive filters are trained with seven true-class templates per target class.

The phase constraints specified for two targets are $\phi^1 = \pi/4$ ($45^\circ$) for target 1 and $\phi^2 = \pi/2$ ($90^\circ$) for target 2. The goal DC specified for the training process is $DC_g = 0.8$. Fig. 9 shows the output correlation intensity planes obtained with filters 28 and 12 of the filter bank for the input scene shown in Fig. 8(e). Observe that each plane contains a sharp peak indicating the presence of one target at the correct position. Also, we see that the output-correlation intensity values in the background area are very low in each plane. Table 1 shows the performance of two AMSDF filters used in Fig. 9. We see that two targets are recognized with a DC value of about 0.75 which is close to the goal DC used for training. Also, observe that resultant phase values $\phi^1$ and $\phi^2$ evaluated at coordinates of the correlation peak in each plane are close to the phase values previously specified in the filter constraints. This means that the two targets can be recognized and correctly classified with a bank of AMSDF filters. Furthermore, observe that since only filters 12 and 28 of the bank can yield high DC values in response to the input frame, we easily estimate that the rotation angle of target 1 is in the sector of $270^\circ \pm 5^\circ$ whereas the rotation angle of target 2 is in the sector of $110^\circ \pm 5^\circ$. In general, a sector for a orientation angle estimate is obtained as $\theta_k = 2\phi_k(f_k(k)-1) \pm \epsilon_\theta$ degrees, where $f_k$ is the index of the filter in the bank which yields the highest DC value for $k$th frame. Note that these sectors can be used in prediction stage to choose the required filters to perform accurate estimates of targets states in consequent frame.

Now we evaluate the performance of all considered tracking systems in 400 scene frames for different additive noise SNR values. The proposed system operates accordingly to the algorithm shown in Fig. 5. A filter bank containing 36 AMSDF filters trained to recognize and classify two targets with tolerance to in-plane rotations within the range of [0, 360]$^\circ$ was designed. The phase constraints set for two target classes are $\phi^1 = \pi/4$ ($45^\circ$) and $\phi^2 = \pi/2$ ($90^\circ$). The goal DC used for the training process is $DC_g = 0.9$. To estimate the position of targets we use the MAP estimator given in Eq. (20). The a-priori probability distribution for a target location is modeled as follows:

\[
P(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left(-\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right),
\]

where $(x_0, y_0)$ are predicted coordinates of location of the $q$th target, and $\sigma_x$ and $\sigma_y$ represent motion deviations in $(x, y)$ directions computed for each frame as $\sigma_x = x_k^q - x_{k-1}^q$, $\sigma_y = y_k^q - y_{k-1}^q$.

The MACH-PDCCF system utilizes two filter banks; one of them is used for detection of two targets while another bank is used to classify detected objects [31,32]. The detection bank contains 36 MACH filters per target class (72 filters in total). Each MACH filter is trained to recognize one target with a specific rotation angle $\theta$ with tolerance of $\pm 5^\circ$. All MACH filters are trained using eleven true-class templates and eleven false-class templates. Also, the filters use the image of the background as an additional false-class template to be rejected. The PDCC filter bank (classification bank) contains 36 PDCC filters. Each PDCC filter is trained with eleven templates per target class. The training templates used to synthesize the PDCC filters are cropped to 60 x 60 pixels [31]. Powers of 1.5 and 2 for required nonlinear transformation of input fragments in PDCC formulation are used. In this system, an incoming scene frame is correlated with all 72 MACH filters of the detection bank.

**Fig. 10.** DC performance with 95% confidence of tested systems in 400 scene frames. (a) Target class 1. (b) Target class 2. (c) Absence of targets (background and additive noise only).
Afterwards, current scene frame is fragmented into several subframes of $60 \times 60$ pixels, each of them created around the coordinates of a correlation peak with the higher DC values among all correlation planes. It is expected that each subframe contains at least one target of unknown class. Next, the subframes are processed with a PDCC filter bank to establish the class for each detected target. This procedure is repeated for all frames. In this approach, an estimate of an orientation state for a target is taken as the coordinates of the correlation peak with the highest DC value in all planes from the MACH filter bank. An estimate of an orientation angle for a $q$th target is calculated as $\theta_q^k = 10 \left( r_q^k - 1 \right)$, where $r_q^k$ is the index of a filter in the bank of $q$th class that yields the highest DC value for $k$th frame.

The system proposed by Manzur et al. \[6\] employs 180 BPO filters per target class. Each BPO filter is designed to recognize one target with a specific orientation angle $\theta$ with tolerance of $\pm 1^\circ$. This system uses a total of 360 BPO correlation filters. In this system, an incoming frame is correlated with 180 BPO filters for target class 1 and with 180 BPO filters for target class 2. The correlation intensity plane with the highest DC value for each target class is chosen for searching the correlation peaks. The coordinates of the correlation peak with the highest DC value in each target class are taken as an estimate of the position state of the target in each frame. Moreover, an estimate of an orientation state for each detected target is calculated as $\theta_q^k = 2 \left( r_q^k - 1 \right)$. Note that in this approach the estimated sectors for orientation angles are narrower than those sectors obtained with proposed and MACH-PDCC systems.

With 95% confidence the results for all tested systems given in terms of the DC are presented in Fig. 10. Observe that proposed system (referred to as AMSDF) yields the best results. Also note that proposed system is robust to additive noise because it is able to yield good DC values for target detection even in highly noisy conditions of 0 dB SNR. The MACH-PDCC system yields similar results than those obtained with the proposed system only for target 2. However for target 1 the results obtained with proposed system are significantly better. It is important to observe, that difficulties in detecting target 1 for MACH-PDCC and BPO systems are due to because the gray contents of target 1 is very similar to the local contents of the background in several regions. Thus, MACH and BPO filters are unable to distinguish properly among target 1 and the background in regions having similar structures. As result, we can conclude that proposed adaptive approach used for filter design is more robust to additive overlapped and nonstationary disjoint noise than MACH and BPO filter designs. The BPO-based system is very sensitive to additive noise. It can be seen how for 0 dB SNR the performance of the BPO system (for two targets) is considerably lower than the performance of the rest tested systems.

Now, we evaluate the classification performance of all tested systems in 400 scene frames. The results are summarized in Tables 2–4. The entries in Tables 2–4 show the number of input objects of each target class (denominators) versus the correct decisions made by the systems in 400 frames (denominators). The decision for target detection with tested systems is taken as follows: first, the highest DC value is chosen and then it is compared with a predefined threshold. If a target is present in the scene, only false alarms may occur. The DC is able to well control these errors. The adaptive filter output contains a well-defined maximum and the DC $> 0$ means no false alarms (see Fig. 10(a) and (b)). If a target is absent in the scene, miss errors may occur. In this case the value of DC is important. The system output has no obvious global maximum in this situation. So, the DC value computed with two local maxima (one for correlation peak and the second one for sidelobe) is relatively low (see Fig. 10(a) and (b)). Therefore, to take into account two kinds of errors we can make decision based on the DC value with thresholding. For all tested systems with 95% confidence the threshold can be chosen as a function of the input SNR using Fig. 10(a) and (b). For instance, for the proposed system threshold values for input SNR values of 100 dB, 20 dB, 10 dB, and 0 dB can be chosen from the following intervals: $[0.05–0.85]$, $[0.25–0.8]$, $[0.3–0.8]$, and $[0.35–0.7]$, 

### Table 2

Classification performance of proposed AMSDF system.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>100 dB SNR inputs</th>
<th>20 dB SNR inputs</th>
<th>10 dB SNR inputs</th>
<th>0 dB SNR inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td>T1</td>
<td>T2</td>
<td>T1</td>
<td>T2</td>
</tr>
<tr>
<td>T1</td>
<td>400/400</td>
<td>0</td>
<td>400/400</td>
<td>0</td>
</tr>
<tr>
<td>T2</td>
<td>0</td>
<td>400/400</td>
<td>0</td>
<td>400/400</td>
</tr>
<tr>
<td>Reject</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ERC</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

### Table 3

Classification performance of MACH-PDCC system.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>100 dB SNR inputs</th>
<th>20 dB SNR inputs</th>
<th>10 dB SNR inputs</th>
<th>0 dB SNR inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td>T1</td>
<td>T2</td>
<td>T1</td>
<td>T2</td>
</tr>
<tr>
<td>T1</td>
<td>391/400</td>
<td>0</td>
<td>363/400</td>
<td>0</td>
</tr>
<tr>
<td>T2</td>
<td>4</td>
<td>399/400</td>
<td>7</td>
<td>398/400</td>
</tr>
<tr>
<td>Reject</td>
<td>5</td>
<td>1</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>ERC</td>
<td>2.7%</td>
<td>0.25%</td>
<td>9.2%</td>
<td>0.5%</td>
</tr>
</tbody>
</table>

### Table 4

Classification performance of BPO system.

<table>
<thead>
<tr>
<th>Outputs</th>
<th>100 dB SNR inputs</th>
<th>20 dB SNR inputs</th>
<th>10 dB SNR inputs</th>
<th>0 dB SNR inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision</td>
<td>T1</td>
<td>T2</td>
<td>T1</td>
<td>T2</td>
</tr>
<tr>
<td>T1</td>
<td>400/400</td>
<td>0</td>
<td>383/400</td>
<td>0</td>
</tr>
<tr>
<td>T2</td>
<td>0</td>
<td>400/400</td>
<td>0</td>
<td>384/400</td>
</tr>
<tr>
<td>Reject</td>
<td>0</td>
<td>0</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>ERC</td>
<td>0%</td>
<td>0%</td>
<td>4.2%</td>
<td>4%</td>
</tr>
</tbody>
</table>
respectively. In our experiments for all tests we assigned the threshold value to 0.5. In a similar manner the DC threshold values can be set for other two tested systems. However, when the noise is heavy (0 dB) then there are no ways to select correctly the threshold values (see Fig. 10(a) and (c) for MACH-PDCC and BPO). This is because the DC values for true positive detection are lower than ones for false positive detection. It means either false alarms or miss errors always will occur. The next step is to take classification decision. It can be done by analyzing the phase (correlation plane is complex) at the position of the DC maximum. The phase value belongs to a sector in the complex plane, which indicates the class of the object. In the evaluation stage the LE of a detected object is also considered. If the LE is less than half of the target size (25 pixels), then the object is correctly detected, otherwise, it is supposed that the system yields a wrong detection. Tables 2–4 also show the percentage of error rate classification (ERC), calculated as

$$ERC = \frac{\text{missclassifications} + \text{rejects}}{\text{number of frames}} \times 100.$$  

(26)

We can see from Table 2 that proposed system yields no rejections and no missclassifications in 400 frames for 100 dB and 20 dB additive noise SNR, and yields only one rejection in 400 frames for 10 dB SNR. Also, observe that proposed system yields low ERC percentages of 3% for target 1 and 0.75% for target 2, in highly noisy conditions of 0 dB SNR. This means that proposed system is

![Fig. 11. True and estimated state trajectories for two targets obtained with proposed system in 400 scene frames in environment of 10 dB additive noise SNR. (a) State trajectory for target 1. (b) State trajectory for target 2.](image)

![Fig. 12. True and estimated state trajectories for two targets obtained with MACH-PDCC system in 400 scene frames in environment of 10 dB additive noise SNR. (a) State trajectory for target 1. (b) State trajectory for target 2.](image)

![Fig. 13. True and estimated state trajectories for two targets obtained with BPO-based system in 400 scene frames in environment of 10 dB additive noise SNR. (a) State trajectory for target 1. (b) State trajectory for target 2.](image)
reliable and robust to additive noise. The BPO system, yields 0% ERC for 100 dB SNR (see Table 4). However, the classification performance of this system worsens rapidly when the SNR decreases. Note that BPO-based system yields a very poor performance in environment of 0 dB SNR. The MACH-PDCC system yields the worst ERC percentages due to misclassifications (see Table 3). For 100 dB SNR this system yields four missclassifications and five rejections for target 1. However, when the SNR decreases

the number of rejections and misclassifications for target class 1 increases rapidly. Additionally, observe that MACH-PDCC system yields good results for target 2 in noisy conditions of 100 dB, 20 dB and 10 dB SNR, and yields a ERC percentage of 54.7% for 0 dB SNR. The BPO-based system yields very good results in terms of misclassifications for two targets. Nevertheless, for low SNR values the number of target rejections increases considerably. Also note that for 0 dB SNR the BPO-based system yields very high ERC percentages of 98.7% and 99.2% for target class 1 and target class 2, respectively.

Next, we evaluate the tracking accuracy of all considered systems in 400 scene frames. These results are measured in terms of LE and OE across true $s_k^q$ and estimated $\hat{s}_k^q$ state vectors in all scene frames. Fig. 11 shows the true and estimated state trajectories for two targets in 400 scene frames, when estimated trajectories are obtained with the proposed system in environment of 10 dB additive noise SNR. Observe that there is a good correspondence between true and estimated state trajectories for two targets. Also, observe that there are no false detections (detections occurring in the area of the background) in any scene frames. Additionally, Figs. 12 and 13 show the true and estimated state trajectories for two targets obtained with MACH-PDCC and BPO systems, respectively. From Fig. 12(a), we see that for target 1 the MACH-PDCC system yields several false location estimates which occur in the background area. In contrast, this system yields good results for target 2. Also note from Fig. 13 that BPO-based system yields various orientation state estimates with large errors of about 180°. This large error happen because eventually a BPO filter trained to recognize the target rotated by 180° with respect
to the true orientation angle of the target yields a higher DC value than that obtained with the correct filter. Note that these incorrect estimates will lead to large LE and OE values. The results with 95% confidence given in terms of LE and OE are for all tested systems presented in Figs. 14 and 15, respectively. From Fig. 14 we can see that proposed system yields the best results in terms of LE in majority of the tests. The MACH-PDCC system yields very small LE values for target 2 in environments of 100 dB, 20 dB and 10 dB SNR. However, the performance of this system in terms of LE is rapidly reduced for 0 dB SNR. Furthermore, note that for target 1 the MACH-PDCC system yields a poor performance in terms of LE. This poor performance is mainly due to false detections occurring in the background area (see Fig. 12(a)). Actually, observe that for target 2 the LE performance of MACH-PDCC system is slightly better than the LE performance obtained with proposed approach. This behavior coincides with state trajectories depicted in Fig. 12(a). The BPO-based system yields an acceptable LE performance only for low noise of 100 dB and 20 dB SNR. The LE performance of the BPO-based system decreases drastically in highly noisy conditions of 10 dB and 0 dB SNR. Next from Fig. 15 we can see that proposed AMSDF system yields the best results in terms of OE in majority of cases.

The proposed system is able to estimate the orientation angle of two targets with a good precision even in highly noisy conditions. The MACH-PDCC approach also yields good results in terms of OE for two targets, with the exception of 0 dB SNR. The BPO-based system yields very poor results in all the cases. This is because as shown in Fig. 13 the BPO system is unable to distinguish reliably a target rotated by 180° with respect to the true orientation angle of the target.

Now we evaluate the speed of processing of all tested systems in 400 scene frames. Our implementations were carried out in MATLAB with commercial toolbox JACKET to enable GPU processing. JACKET is a software platform developed at AccelerEyes [46] that provides full GPU computational capability for Compute Unified Device Architecture (CUDA) hardware [47] such as NVIDIA GPUs. Tracking systems were tested on a laptop computer with 2 GHz Intel CORE i7 CPU, 8 MB of RAM memory, and a NVIDIA GeForce GTX560M GPU with 1.5 MB of memory. The filter banks for each tracking system were implemented on the GPU in blocks of 36 correlation filters in parallel. We calculate the average frames-per-second for each system in different environments of additive noise SNR. The obtained results are presented in Fig. 16. Note that proposed system achieves speed rates above 30 frames-per-second in all the tests. This indicates that proposed system is suitable for real-time applications by exploiting massive parallelism in digital hardware. In contrast, note that MACH-PDCC and BPO systems operate at low speed rates below 5 frames-per-second. Consequently, these systems can only be implemented in hybrid opto-digital correlators in order to achieve real-time operation.

Fig. 17. Optical setup for real images.

Fig. 18. Examples of real scene frames captured with the optical setup shown in Fig. 17.
Finally, we test the proposed system in real images. The optical setup used for this experiment is shown in Fig. 17. We record a sequence of 100 scene frames containing two targets that move in horizontal plane over the background at unknown coordinates. Fig. 18 shows examples of scene frames processed with the proposed system. One can observe that the system is able to locate and classify two targets in all frames. Fig. 18 shows estimated tracks of the state of the targets in each frame. A high precision of estimated location and orientation states of two targets in all frames are obtained.

5. Conclusions

A real-time system for detection, classification and tracking of multiple moving objects in noisy scenes was presented. The proposed system employs a filter bank of adaptive multiclass composite filters with complex constraints to estimate in real-time the state trajectory of various moving targets in a sequence of images. The proposed system employs a prediction stage based on modeling the kinematic behavior of targets in two-dimensional space. Based on predicted states of each target the system splits the input frame into several small subframes to perform accurate and fast state estimates by processing only the small subframes. Moreover, by using predicted orientation states of the targets in current frame the system adjust the required number of filters in the bank to reduce the number of correlations and, therefore, to increase the speed of processing of the overall system. According with computer simulation results the proposed system showed a superior performance in terms of target detection and classification, tracking accuracy and speed of processing comparing with recent state of the art tracking systems based on correlation filtering. Also, computer simulations results showed that proposed system is robust to overlapped and disjoint noise. The proposed system was implemented in a GPU by exploiting massive parallelism. Accordingly with test speed carried out for tracking systems the proposed system yields processing rates above 30 frames-per-second which is suitable for real-time applications using digital hardware. Finally, proposed system was tested using real-life images. According to obtained results, the proposed system can be used for real-time tracking applications using digital hardware.

Acknowledgements

This research was supported by Consejo Nacional de Ciencia y Tecnología (CONACYT) and Secretaría de Investigación y Posgrado - Instituto Politecnico Nacional (SIP-IPN), through the projects CONACYT-0130504 and SIP20120310.

References