Multiframe distortion-tolerant correlation filtering for video sequences

R. Kerekes, B. Narayanaswamy, M. Beattie, B. V. K. Vijaya Kumar, and M. Savvides

Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT

Distortion-tolerant correlation filter methods have been applied to many video-based automatic target recognition (ATR) applications, but in a single-frame architecture. In this paper we introduce an efficient framework for combining information from multiple correlation outputs in a probabilistic way. Our framework is capable of handling scenes with an unknown number of targets at unknown positions. The main algorithm in our framework uses a probabilistic mapping of the correlation outputs and takes advantage of a position-independent target motion model in order to efficiently compute posterior target location probabilities. An important feature of the framework is the ability to incorporate any existing correlation filter design, thus facilitating the construction of a distortion-tolerant multi-frame ATR. In our simulations, we incorporate the minimum average correlation energy Mellin radial harmonic (MACE-MRH) correlation filter design, which allows the user to specify the desired scale response of the filter. We test our algorithm on real and synthesized infrared (IR) video sequences that exhibit various degrees of target scale distortion. Our simulation results show that the multi-frame algorithm significantly improves the recognition performance of a MACE-MRH filter while requiring only a marginal increase in computation. We also show that, for an equivalent amount of added computation, using larger filter banks instead of multi-frame information is unable to provide a comparable performance increase.

1. INTRODUCTION

Distortion-tolerant correlation filters (CFs) have proven to be successful for the task of target detection and recognition on single-frame images[1] due to their attractive properties such as shift-invariance and graceful degradation. Because of these properties, they are well-suited for processing scenes with an unknown number of targets and large amounts of noise and/or clutter. For applications in which a time sequence of images of the scene is available, one approach is to apply a single-frame recognition algorithm to each frame separately and to declare the presence of a target if a majority of the frames detect the target of interest. While the implementation of such a system is a straightforward extension of a single-frame classifier, it does not make use of the positional continuity of a moving target across frames. A better strategy might recognize that the target is not likely to move randomly from one region of the image to another region far from it, and hence the target location extracted from frame n might make a good first estimate of the target location in frame $n + 1$. Such a strategy could be used to improve target recognition by augmenting the result in a particular frame with information gathered from previous frames and by imposing a stochastic model for how the target might move from one frame to another.

Target recognition across multiple frames in this manner is sometimes referred to as tracking. Several solutions have been proposed in the literature. The most obvious solutions combine correlation filtering methods with some variety of Kalman-Bucy filters (KBfs).[2, 3] The typical approach of such solutions is to apply the Kalman filter to initial estimates of the target locations generated by the correlation filters. Such methods have the disadvantage of tracking only a single potential target, since they assume unimodal Gaussian posteriors. Kalman filters are also known to suffer from the problem of break-off; that is, if the tracker locks onto the wrong target, it has very little chance of recovery. This suboptimal coupling of KBf trackers and CFs has exhibited poor performance in scenes with large amounts of clutter. [4]

Bruno has proposed Bayesian methods for target tracking based on hidden Markov models (HMMs) which overcome several of the limitations of KBf/correlation filter coupling.[5] While this solution derives an optimal estimate for the position of a target based on past and future frames, it nonetheless suffers from several drawbacks, including high computational load, which increases with the size and number of models of the target, and the inability to track multiple targets simultaneously. The use of particle filters was proposed by the same author as a way to decrease computation and move to a continuous-valued target motion model. Particle filters, unlike Kalman filters, are not limited to unimodal posteriors; however, such techniques have not been applied to correlation filter-based ATR. Furthermore, this departure from the optimal HMM solution results in a loss of performance.

Several other solutions to the tracking problem have also been proposed in the literature. In Arnold *et al.*,[6] dynamic programming is employed to find the most likely path of a target across a sequence of frames. For scenes with a large number of potential target locations (possibly every pixel of the scene), this procedure can become computationally challenging. In Lipton *et al.*,[7] detection is applied only to the most recently observed target location and any image regions that change from one frame to the next. This approach enables more efficient tracking, but may not be effective in scenes containing extensive non-target motion.

While distortion-tolerant CFs have proven useful as single-frame classifiers, they present no obvious strategy for tracking across multiple frames apart from introducing an intermediate detection step, which results in a loss of information. However, if the resulting correlation output values are placed in a probabilistic framework, information from multiple frames can be combined in a Bayesian manner. In this paper, we present a strategy somewhat related to the solution proposed in[5] that offers certain computational advantages over a traditional HMM-based approach. We begin by imposing a probabilistic interpretation of the CF outputs. We then process these outputs in a way analogous to the HMM forward algorithm[8] to generate an enhanced correlation output based on the outputs from previous frames. We show how certain reasonable assumptions on the motion model can be used to structure the forward algorithm as a convolution, thus making our algorithm require only two additional fast Fourier transforms (FFTs) per frame. We use a variant of the recently-introduced Minimum Average Correlation Energy Mellin Radial Harmonic (MACE-MRH) filters to achieve scale-tolerant target recognition. Our simulation results show that while the single frame performance of these filters is often poor in our test scenarios, coupling the filters with the multi-frame algorithm enables high accuracy target recognition while retaining scale-tolerance.

The rest of this paper is organized as follows. Section 2 provides a brief background on MACE-MRH correlation filters. Section 3 explains the theory behind the multi-frame algorithm and the assumptions made on the target motion model. Section 4 describes our implementation and shows several experimental results, and we provide concluding remarks in Section 5.

2. SCALE-TOLERANT CORRELATION FILTERS

Correlation filtering refers to the process of locating a specific pattern in an image by computing its cross-correlation with a filter template. The resulting output, called the "correlation plane," is inspected for peaks, and the locations of sufficiently sharp peaks indicate the positions of the pattern in the input. Thus, CFs can handle the presence of multiple targets in the scene. Peak sharpness is typically measured by the peak-to-sidelobe ratio (PSR) defined[9] as follows:

$$PSR = \frac{|peak - mean|}{std} \tag{1}$$

where the mean and standard deviation are computed from a small region of the plane surrounding but not including the peak.

The simplest CF is the matched filter in which the template is matched to a single training image; however, the use of matched filters is typically not attractive for ATR since the number of matched filters needed may be very large because of target variability. Such variability is often due to distortions such as scale, rotation, configuration, and thermal state. In contrast, more advanced composite CF designs allow the use of multiple training images and can produce a template that tolerates one or more types of distortion. Many designs also optimize certain performance criteria such as peak sharpness, output similarity, and low output noise variance.[10] Thus, we will need fewer composite CFs than matched filters. The design stage of a CF can be computationally intensive; however, the filter need only be designed once, and applying the filter thereafter can be performed efficiently using FFTs.

MACE-MRH correlation filters[11] are specifically formulated to tolerate target scale distortion. These filters are based on the Mellin radial harmonic (MRH) transform, an orthogonal basis set with the property that scaling a signal with respect to the spatial axes only affects its MRH coefficients by a phase factor. This scaling property is exploited in the MACE-MRH filter design theory to yield a filter with a user-controlled scale response; that is, the user can specify the desired correlation peak response curve of the filter over all input scale factors. A typical family of scale response curves is the set of rectangular functions, which allow a given filter to recognize the input pattern within some limited range of scale factors. A rectangular scale response curve is useful in designing banks of scale-tolerant filters, where each filter in the bank is responsible for recognizing targets in some small partition of the total range of scale distortion. MACE-MRH filters have higher discrimination capability than earlier scale-invariant filter designs[12] because they retain much more of the available pattern information.

It has been demonstrated[13] that applying fractional-power nonlinearities in the frequency domain can improve the recognition performance of a MACE-MRH filter. One reason for this improvement is that raising the magnitude of the frequency spectrum to a power between zero and one typically has the effect of boosting the higher, more discriminating frequencies (since these most often have lower magnitude), resulting in a sharper and more detectable correlation peak. We use fractional-power enhancements in our experiments, and thus we refer to our overall filtering scheme as fractional-power Mellin radial harmonic (FPMRH) correlation filtering.

In an application where the scale of the observed target is highly uncertain, we might choose to train a single FPMRH filter with a very wide rectangular response curve on some reference image of the target. Although this approach saves computation, one disadvantage is that enforcing tolerance over a wider range of variations typically leads to decreased peak sharpness and thus lower discrimination capability. In order to achieve a compromise between discrimination and distortion tolerance, multiple filters trained on smaller partitions of the distortion range are often applied in parallel at the cost of increased computation. Thus for each input frame, a group of correlation planes is obtained at the output. In this paper, several of our simulations use only a single filter for each recognition task, and we show that this can be sufficient for accurate target recognition when used in conjunction with the multi-frame algorithm.

3. EFFICIENT MULTI-FRAME CORRELATION FILTERING

Our goal in multi-frame correlation filtering at each time step is to combine the information contained in the correlation planes generated from the current and past frames in order to produce a single, high-accuracy plane that contains a reduced number of false peaks. This can be challenging in ATR applications where correlation filters need to be trained to handle a wide range of target scales and orientations. As a consequence of achieving such distortion tolerance, the filters tend to produce weak correlation peaks that are easily corrupted by noise. By incorporating information from previous planes, however, we may be able to lock onto consistent peaks while rejecting more of the noise and short-lived spurious peaks in the correlation planes.

The key to our multi-frame filtering strategy is a probabilistic interpretation of correlation planes. We first map each individual correlation plane to a "probability plane". This is a 2-D array in which each value represents the probability that a target is located at the corresponding pixel position in the scene. (It is important to note here that when we say that a target is located at a pixel p in an image, we mean that some pre-chosen aim point on the target is located most closely to pixel p; we do *not* mean simply that pixel p is somewhere on the target.) We define another array, called the "transition plane" and denoted by $f_{TP}\left(\Delta\mathbf{x}\right)$, the values of which specify the probabilities that a target will move from its current position by $\Delta\mathbf{x}$ to other possible positions in the scene. From this probabilistic information and a Markovian assumption on the target path, we can compute a plane of values representing the probability of a target appearing at each position in frame n given frames 1 through n.

To formalize this notion, we first apply a mapping function to the raw correlation plane from each frame, and we denote the resultant value at each position by $P\left(H_t\left[\mathbf{x}\right] \mid F_t\right)$ — the probability of the hypothesis $H_t\left[\mathbf{x}\right]$ that a target is present at position $\mathbf{x}$ at time t given just the observation of the frame F_t at that time. It should be noted that, for a given t, $P(\,H_t\left[\mathbf{x}\right]\,)$ is not a probability mass function (pmf) over the variable $\mathbf{x}$ but rather a 2-D array of binary pmfs, i.e., at each pixel location $\mathbf{x}$, there are two possible events (namely, target present or target not present), and their probabilities sum to one. For example, given just the first frame of data, the probability that a target is present at position $\mathbf{x}$ at time t is denoted by $P\left(H_1\left[\mathbf{x}\right] \mid F_1\right)$, and an estimate of this function is obtained by applying the mapping function to the first correlation plane. We assume that the probabilities of jointly observing any two frames are independent conditioned on the target position. Also, we assume a uniform prior $P(\,H_t\left[\mathbf{x}\right]\,) = p_0$ on the target position for every t (before any observations).

The probability after observing the first two frames is then given by

$$\begin{aligned}
P\left(H_2\left[\mathbf{x}\right] \mid F_2, F_1\right) &= \frac{P\left(F_2, F_1 \mid H_2\left[\mathbf{x}\right]\right) P(\,H_2\left[\mathbf{x}\right]\,)}{P(\,F_2, F_1\,)} \\
&= \frac{P\left(F_2 \mid H_2\left[\mathbf{x}\right]\right) P\left(F_1 \mid H_2\left[\mathbf{x}\right]\right) P(\,H_2\left[\mathbf{x}\right]\,)}{P(\,F_2, F_1\,)} \\
&= \frac{P\left(H_2\left[\mathbf{x}\right] \mid F_2\right) P(\,F_2\,)}{P(\,H_2\left[\mathbf{x}\right]\,)} \cdot \frac{P\left(H_2\left[\mathbf{x}\right] \mid F_1\right) P(\,F_1\,)}{P(\,H_2\left[\mathbf{x}\right]\,)} \cdot \frac{P(\,H_2\left[\mathbf{x}\right]\,)}{P(\,F_2, F_1\,)} \\
&= \frac{P\left(H_2\left[\mathbf{x}\right] \mid F_2\right) P\left(H_2\left[\mathbf{x}\right] \mid F_1\right)}{C_2}, \qquad (2)
\end{aligned}$$

where

$$C_2 \equiv P(\,H_2\left[\mathbf{x}\right]\,) \cdot \frac{P(\,F_2, F_1\,)}{P(\,F_2\,)P(\,F_1\,)} = p_0 \cdot \frac{P(\,F_2, F_1\,)}{P(\,F_2\,)P(\,F_1\,)} \qquad (3)$$

corresponds to the second frame and is a constant with respect to $\mathbf{x}$. If we assume that two targets cannot be located at the same position, then we can rewrite Equation 2 as

$$P\left(H_2\left[\mathbf{x}\right] \mid F_2, F_1\right) = \frac{P\left(H_2\left[\mathbf{x}\right] \mid F_2\right) \cdot \sum_{\mathbf{x}'} P\left(H_2\left[\mathbf{x}\right] \mid H_1\left[\mathbf{x}'\right], F_1\right) P\left(H_1\left[\mathbf{x}'\right] \mid F_1\right)}{C_2}. \qquad (4)$$

The vector $\mathbf{x}'$ in Equation 4 is the summaion index, where the summation takes into account all possible locations $\mathbf{x}\,'$ at time 1 from which a target could have arrived at location $\mathbf{x}$ at time 2. We now introduce the assumption that the probability of a target moving from one pixel to another can be specified by a function of a single argument $f_{\,TP}\left(\Delta\mathbf{x}\right)$, i.e., the transition probabilities are spatially stationary, depending only on the change in target position and not on the absolute position itself. Under this assumption and substituting $\mathbf{x} - \mathbf{x}'$ for $\Delta\mathbf{x}$, we can rewrite Equation 4 as

$$P\left(H_2\left[\mathbf{x}\right] \mid F_2, F_1\right) = \frac{P\left(H_2\left[\mathbf{x}\right] \mid F_2\right) \cdot \sum_{\mathbf{x}'} f_{TP}\left(\mathbf{x} - \mathbf{x}'\right) P\left(H_1\left[\mathbf{x}'\right] \mid F_1\right)}{C_2}.$$

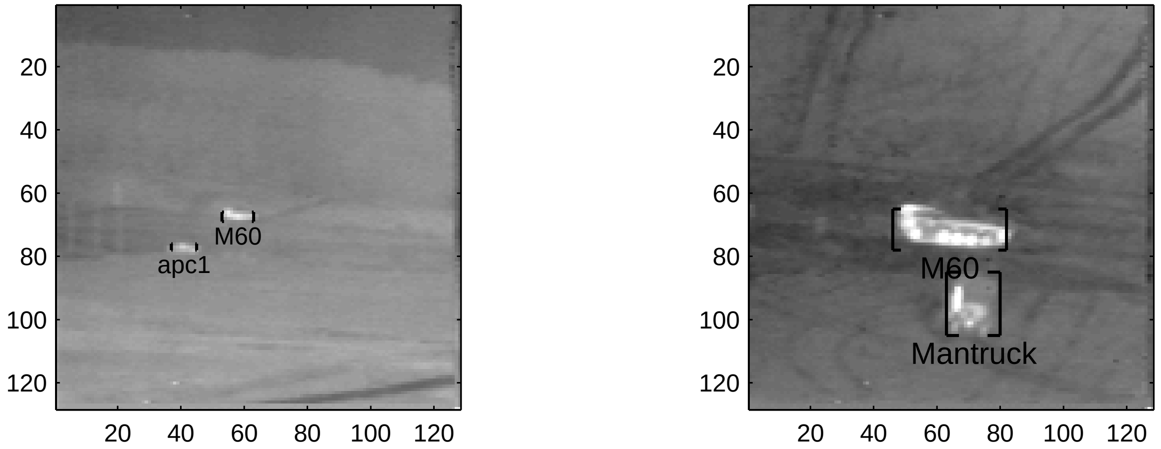
Using these expressions, we can generalize the probability of a target being present at position $\mathbf{x}$ at time t in terms of all previous frames as follows:

$$P\left(H_t\left[\mathbf{x}\right] \mid F_t, F_{t-1}, ..., F_1\right) = \frac{1}{C_t} \cdot P\left(H_t\left[\mathbf{x}\right] \mid F_t\right) \cdot \left[f_{TP}\left(\mathbf{x}\right) * P\left(H_{t-1}\left[\mathbf{x}\right] \mid F_{t-1}, F_{t-2}, ..., F_1\right)\right] \qquad (5)$$

where the $*$ and $\cdot$ operators denote discrete 2-D convolution and pointwise multiplication, respectively. We refer to this resulting plane as the "enhanced plane", because pointwise-multiplying the plane $P\left(H_{\,t}\left[\mathbf{x}\right] \mid F_t\right)$ by the plane resulting from the convolution operation in Equation 5 effectively enhances the original plane by incorporating additional information.

To generate "probability planes" $P\left(H_t\left[\mathbf{x}\right] \mid F_t\right)$ from the raw correlation outputs, we use the PSR metric given in Equation 1. This metric has been shown to be effective for locating targets in cluttered scenes.[1] It is important to realize that PSR values are not limited to the range $[0, 1]$ (and in fact are not bounded above at all). For this reason, they cannot properly be labeled as probability values without first applying some mapping from $[0, \infty]$ to $[0, 1]$. Nevertheless, we choose here to overlook this formality; one justification for this is that when noise is present in the application, PSR values are essentially bounded, and thus there exists a scaling factor which will almost surely map any resultant PSR value into the proper range. However, because this scaling factor will be absorbed by subsequent normalization operations, it is sufficient to work with unscaled PSR values.

While there is no reason to believe that PSR values are directly proportional to the true probability values, they should follow the same trends. In other words, sharp peaks in the correlation plane intuitively correspond to a higher likelihood of the presence of a target, and the PSR metric emphasizes such locations. In our PSR computations, we used a main window size of 20x20 pixels centered around the peak and excluded a 6x6 window immediately adjacent to the peak. For the transition function, we assume a two-dimensional circularly-symmetric Gaussian centered at the origin (where the origin corresponds to no movement). The standard deviation or width $\sigma_{\,\mathrm{TP}}$ of this Gaussian is an important parameter in



(a)

(b)

Figure 1. Frames of sequence “L1608” (ground truth drawn): (a) frame 1; (b) frame 290.

our multi-frame algorithm. If σ_{TP} is close to zero, then the multi-frame algorithm will favor only small motions between adjacent frames. On the other hand, if σ_{TP} is very large, then transition probabilities are effectively uniform over the region of interest and no target location information will be propagated from previous frames to the current frame. We discuss the choice of an appropriate σ_{TP} value in more detail in Section 4.

It should be noted that Equation 5 is the convolution of the transition probability with the probability plane constructed from the sequence of previous frames. This observation enables efficient computation in the frequency domain. We point out that by applying 2D FFTs to $f_{TP}\left(\Delta\mathbf{x}\right)$ and $P\left(H_{t-1}\left[\mathbf{x}\right] \mid F_{t-1}, F_{t-2}, ..., F_1\right)$, we can compute the convolution in the frequency domain as a multiplication. We then apply an inverse FFT to convert the result back into the space domain. This is a far less expensive operation than performing the convolution operation in the space domain directly. Because the FFT of $f_{TP}\left(\Delta\mathbf{x}\right)$ need only be computed once, the incorporation of our multi-frame algorithm in a correlation filtering scheme requires only two additional FFTs and one pointwise array multiplication per frame.

4. EVALUATION

In order to evaluate our algorithm, we carried out target recognition experiments on real and synthesized infrared (IR) video sequences. The real sequence is taken from a database of forward-looking infrared (FLIR) imagery provided by the U.S. Army Aviation and Missile Command (AMCOM). This sequence (labeled “L1608”) was captured from a missile seeker camera attached to a helicopter flying an approach trajectory toward several ground targets. Because of this motion path, the targets in the sequence undergo large scale variations over the duration of the sequence. The AMCOM sequence includes ground truth data for every frame, specifying the position, size, and type of all targets appearing in each frame. Sample frames of this sequence are shown in Figure 1.

Each frame of the synthesized sequences was generated by superimposing a cropped IR target image extracted from the AMCOM database sequence over a synthesized background composed of correlated Gaussian noise. White Gaussian noise was then added to the overall image to yield an effective SNR of -15 dB relative to the target-and-background-only image. In one half of the synthesized sequences, the target size was chosen randomly from the range $[50\%, 100\%]$ of original target size, while in the other half the target size was fixed to its original size. A target path for each sequence was computed as a 2-D random walk, with the magnitude of each successive step drawn according to either a Gaussian, exponential, or uniform distribution and the direction of the step drawn from a uniform distribution over $[0, 2\pi)$. This path model may be less predictable and hence more difficult than a real-world target path, which helps to demonstrate the robustness of our algorithm. Sample frames of a synthesized sequence as well as a sample target path are shown in Figure 2.

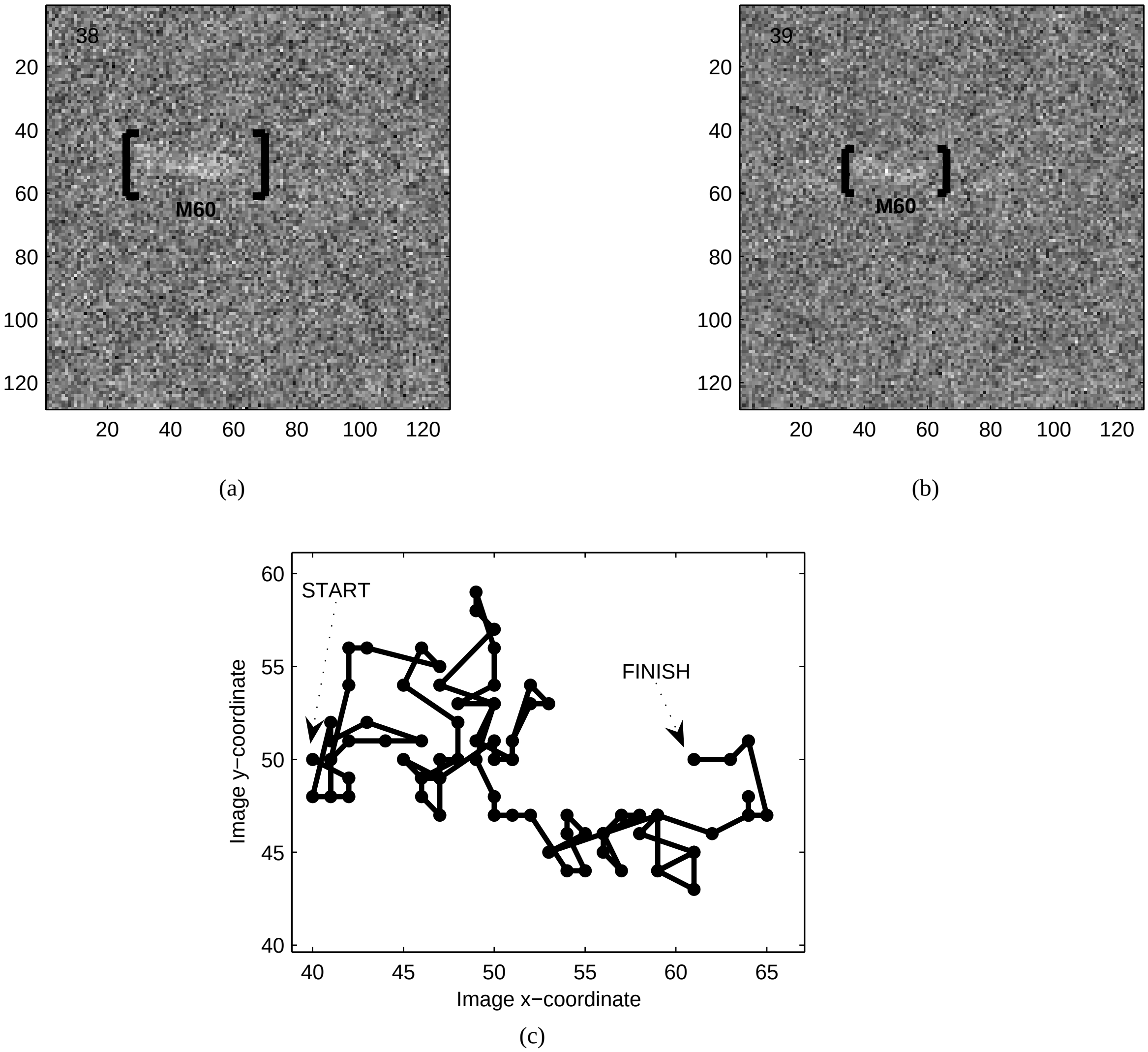
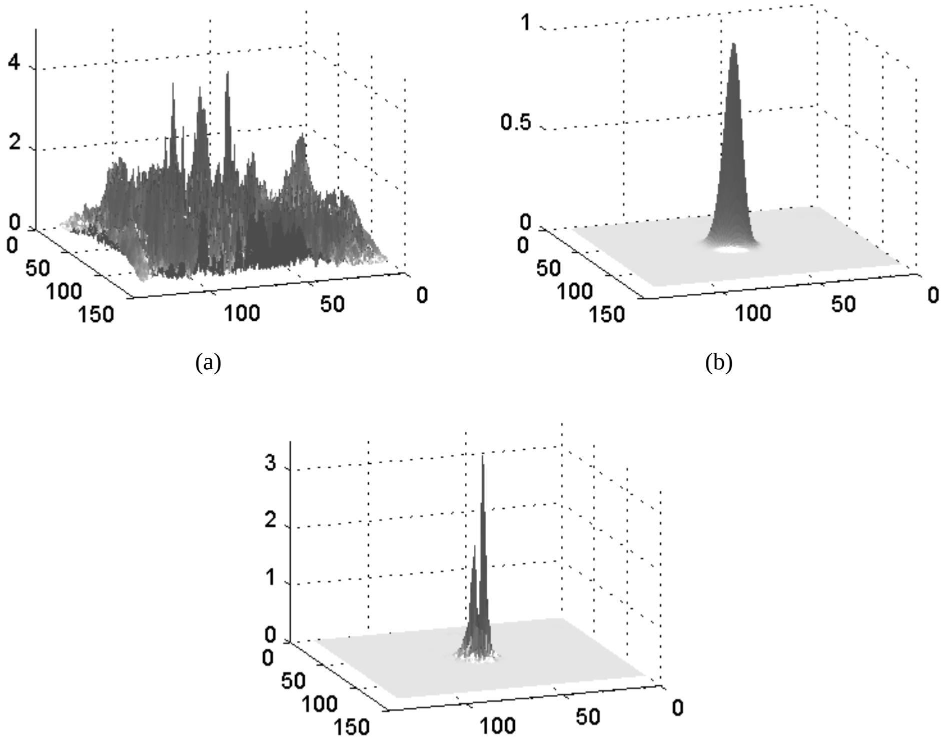


Figure 2. Frames of a synthesized sequence with scale variation (ground truth drawn): (a) frame 38; (b) frame 39. A sample target path from a synthesized sequence is shown in (c).

We trained banks of FPMRH correlation filters to recognize one particular target in each sequence by extracting an image of the target from the final frame of the corresponding AMCOM database sequence. This scheme mimics a realistic scenario in which very few images of the target are available for training during the design stage. In order to train filters with the appropriate scale tolerance, we partitioned the scale range of interest into N non-overlapping subsets, where N is the number of filters in the filter bank, and trained each filter to recognize targets only within its corresponding subset of the scale range. The resulting bank of filters covered the entire range of target scale in the video sequence. As mentioned in Section 2, the advantage of increasing the number of filters in our experiments is that each filter is given a smaller range of scales to tolerate, resulting in sharper correlation peaks and hence better discrimination.

Once the filter bank was designed, we applied it to the video sequence using two different methods: the multi-frame algorithm described in Section 3 (with each bank containing a single filter) and a standard single-frame algorithm. The single-frame algorithm was implemented as follows: first, each of the N correlation filters in the bank was applied to a frame to yield N separate correlation planes. The PSR value was computed at every point in each plane, and a universal



(a)

(b)

(c)

Figure 3. Examples of (a) probability plane, (b) convolution result, and (c) enhanced plane, which is the pointwise product of the two planes in (a) and (b). The planes were generated from the first 30 frames of sequence "L1608" (i.e., at the 30th iteration of the multi-frame algorithm.)

threshold was applied to each PSR plane to yield a set of detections for that plane. The detections from all planes were collected to yield an overall set of detections for the given frame. PSR values were computed using a total of 4 FFTs per plane (for efficiently computing the required first- and second-order statistics by convolving the plane and its pointwise square with window functions), resulting in a total of 5 FFTs per filter used for the single-frame algorithm.

In the multi-frame algorithm, enhanced probability planes were computed from PSR planes as described in Section 3. In order to negate the effect of the unknown constant in Equation 5, the PSR metric was also applied to the resultant enhanced plane. Examples of these planes at each step are shown in Figure 3. The first PSR plane was used as the initial probability plane. The enhanced plane for each frame was then thresholded to yield a set of detections. Our scoring scheme in both the single-frame and the multi-frame method was as follows: first, a successful detection was tallied for each frame in which any detection occurred at a target pixel, and a miss was tallied otherwise; second, a false alarm was tallied for each frame in which any detections occurred outside of the target bounding box. We emphasize the fact that in this scheme, a particular frame can simultaneously produce a detection/false alarm pair or a miss/false alarm pair. False alarm rate and miss rate are denoted by $\mathrm{P}_{\mathrm{FA}}$and P_{M}, respectively.

Our primary analysis is a performance comparison between the multi-frame algorithm and the single-frame approach described above. In order to put the two algorithms on equal footing with respect to computation, we trained filter banks of various sizes for use by the single-frame algorithm. Specifically, a single-frame approach using 2 filters with the PSR metric will require approximately the same amount of computation as the multi-frame algorithm using a single filter. In addition to 2-filter banks, we also trained banks of between 1 and 10 filters, where the larger banks require significantly more computation than the multi-frame algorithm. The performance of the various approaches is shown using box-and-whisker plots in Figure 4 for each type of synthetic sequence that was generated. Each plot represents the false alarm rate statistics over 20 different synthesized sequences. We observe that using a Gaussian-shaped transition probability results in good performance even when the underlying distribution is not Gaussian. Thus, the shape of the transition probability function is not necessarily a critical design choice when using the multi-frame algorithm.

The performance of various filter banks on real data for both the single-frame and the multi-frame algorithms is compared in Figure 5. While intuition suggests that using more filters should increase performance (at the expense of computational load), and indeed this general trend is observed in our results, the multi-frame algorithm with only one filter nevertheless outperforms the single-frame algorithm with as many as 10 filters. Because the multi-frame architecture requires only 2 additional FFTs per frame plus PSR computation, this set of results illustrates the usefulness of incorporating multi-frame information into a correlation filtering scheme.

Our secondary analysis of the multi-frame algorithm explores the effect of the σ_{TP} parameter on recognition performance. For this analysis, we focus on the real sequence "L1608". We first generated receiver operating characteristic (ROC) curves over a wide range of σ_{TP} values and computed the false alarm rate at various fixed miss rates for each curve. Several such ROC curves are shown in Figure 6. It should be noted that using very large σ_{TP} values (> 100) essentially results in a single-frame algorithm; the reason for this is that the flat shape of the resulting Gaussian completely blurs out any information from previous correlation planes in the convolution.

A plot of the false alarm rates versus σ_{TP} for the real sequence is shown in Figure 7. The plot compares single-frame and multi-frame performance as the number of filters in the single-frame filter bank is varied. We first observe that σ_{TP} values close to 5 yield very low false alarm rates (close to zero) compared to the equal-computation (i.e., 2-filter) single-frame case. We also observe a surprising trend in the false alarm rate curve—for σ_{TP} values less than 4 and between 7 and 11, the multi-frame algorithm actually *degrades* recognition performance. For σ_{TP} values in the former range, the transition probability function is too narrow to capture the large motion of the target, and thus the true target peaks are suppressed. On the other hand, we have observed that, for σ_{TP} values in the latter range, the multi-frame algorithm locks on to regions of densely distributed sporadic clutter peaks in the correlation plane. When the average spacing between these disjoint peaks is consistently small relative to the width of the Gaussian transition function, they will tend to reinforce each other in the convolution operation, and the result may outweigh the true peak. If σ_{TP} is in the appropriate range based on the target motion in the sequence, these false peaks will instead will be suppressed by the more consistent true peaks.

5. CONCLUSIONS

We have presented an efficient multi-frame correlation filtering method that merges information from a time sequence of correlation filter outputs using a simple motion model for the targets. An appropriate combination scheme is derived by interpreting a processed version of the correlation plane for each frame as the probability of a target appearing in a particular pixel location. Because this combination of information can be written as a convolution operation, the algorithm can be carried out efficiently in the frequency domain.

Evaluation against a collection of both real and synthesized infrared image sequences indicates that this procedure can be useful for improving target recognition accuracy in practical applications. In all of our test cases, applying the multi-frame algorithm with an appropriate σ_{TP} value resulted in a greater performance improvement than adding an additional filter—which represents an equivalent increase in computation in our implementation. In many of the test sequences, a single FPMRH filter alone produced a false alarm in every frame at a 90% detection rate, while the same filter combined with the multi-frame algorithm at the same detection rate was able to achieve false alarm rates of less than 5%. These large performance increases demonstrate the fact that correlation filters which have high distortion tolerance but poor recognition capability on their own may be useful in scenarios in which information from multiple images is available.

The major parameter to be chosen in the multi-frame algorithm is the width parameter σ_{TP} of a Gaussian transition probability. In addition to demonstrating the existence of beneficial σ_{TP} values, our results suggested that choosing the wrong value of σ_{TP} can have a negative impact on recognition performance. While our analysis involved an extensive search of σ_{TP} values, it did not include a study of the actual target motion statistics in the video sequences. Any reasonable method for choosing good σ_{TP} values would likely require this type of information, and thus future work on our algorithm should include relating σ_{TP} to these statistics. In addition, other classes of transition probability functions and probability mapping functions should be explored, as there is no reason to believe that our preliminary choices are the optimal ones.

6. ACKNOWLEDGEMENTS

The authors would like to thank Dr. S. Richard F. Sims for providing the video data used for testing.

REFERENCES

1. S. R. F. Sims and A. Mahalanobis. Performance evaluation of quadratic correlation filters for target detection and discrimination in infrared imagery. *Optical Engineering*, 43:1705–1711, 2004.
2. R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Engineering*, 82:34–45, 1960.
3. P. S. Maybeck. R. L. Jensen D. A. Hernly. An adaptive extended kalman filter for target image tracking. *IEEE Transactions On Aerospace And Electronic Systems*, 1981.
4. M. G. S. Bruno and J. M. F. Moura. Multiframe detector/tracker: optimal performance. *IEEE Trans. Aero. Elec. Sys.*, 37(3):925–945, 2001.
5. M. G. S. Bruno. Bayesian methods for multiaspect target tracking in image sequences. *IEEE Trans. Signal Processing*, 52(7):1848–1861, 2004.
6. J. Arnold and H. Pasternack. Detection and tracking of low-observable targets through dynamic programming. In *Signal and data processing of small targets 1990; Proceedings of the Meeting, Orlando, FL, Apr. 16-18, 1990 (A91-36901 15-32). Bellingham, WA, Society of Photo-Optical Instrumentation Engineers, 1990.*, pages 207–217, October 1990.
7. A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *WACV '98: Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, page 8, Washington, DC, USA, 1998. IEEE Computer Society.
8. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
9. M. Savvides and B. V. K. Vijaya Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. *IEEE Conference on Advanced Video and Signal Based Surveillance*, 21-23:45–52, 2003.
10. B. V. K. Vijaya Kumar. Tutorial survey of composite filter designs for optical correlators. *Applied Optics*, 31:4473–4481, 1992.
11. R. Kerekes and B. V. K. Vijaya Kumar. Correlation filters with controlled scale response. *IEEE Trans. Image Processing*, to appear.
12. G. Ravichandran and D. Casasent. Advanced in-plane rotation-invariant correlation filters. *IEEE Trans. Pattern Anal. and Machine Intell.*, 16(4):415–420, 1994.
13. R. Kerekes, M. Savvides, B. V. K. Vijaya Kumar, and S. R. F. Sims. Fractional power scale-tolerant correlation filters for enhanced automatic target recognition (atr) performance. *Proc. of SPIE*, 5807:317–328, 2005.

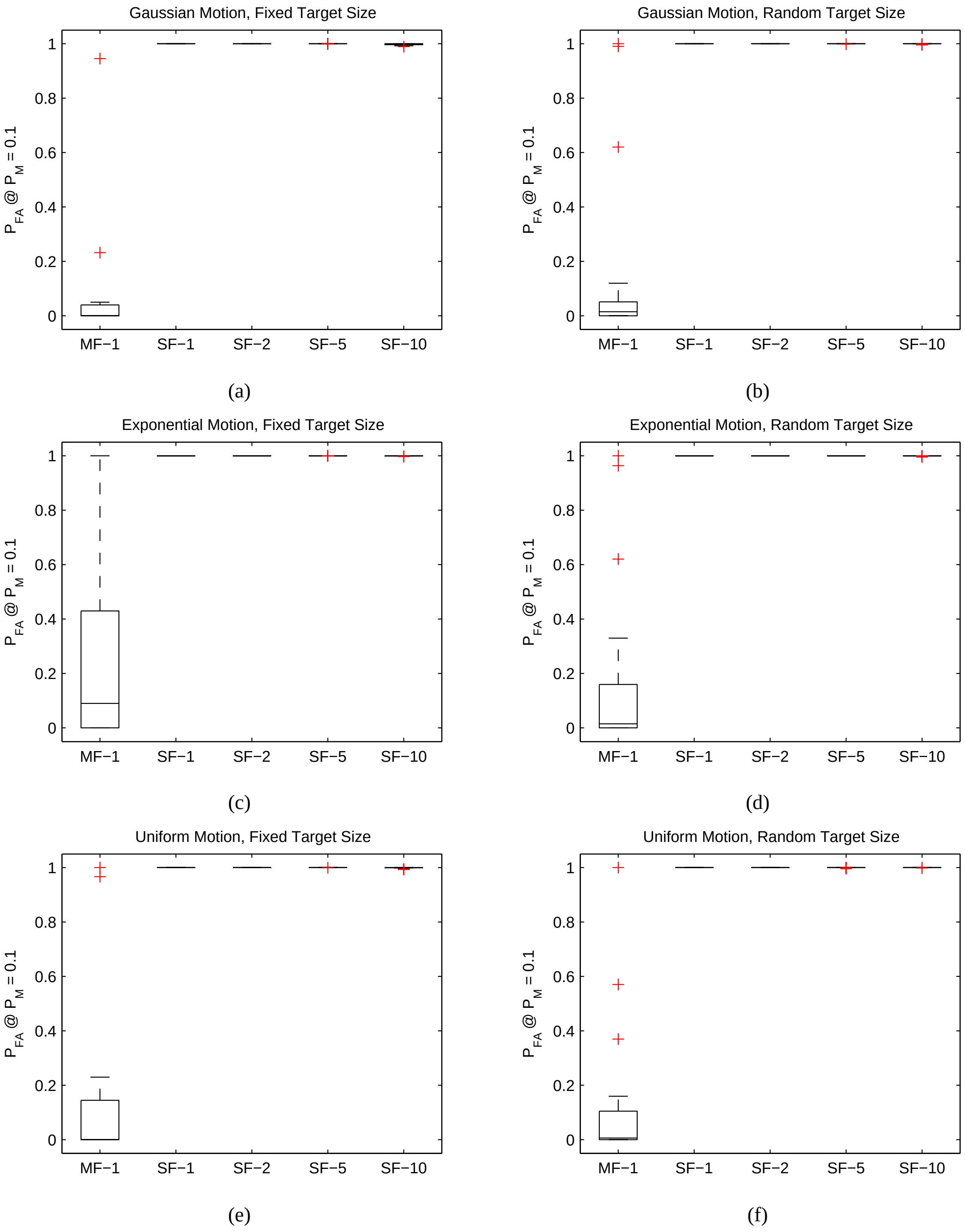


Figure 4. Box-and-whisker plots of false-alarm rates at $\mathrm{P}_{\mathrm{M}} = 0.1$ for multi-frame (MF) and single-frame (SF) recognition on different sets of 20 synthetic sequences. Multi-frame experiments used a single filter, while single-frame experiments used filter banks of sizes 1, 2, 5, and 10. Plots (a), (c), and (e) are for sequences with variable target scale, while plots (b), (d), and (f) have a fixed target size. The title of each plot shows the distribution used to generate target paths in the sequences. Boxes show mean plus lower and upper quartiles. Plus (+) symbols indicate outlier results. Note that the boxes in the single-frame columns are short and appear near the top of the plot.

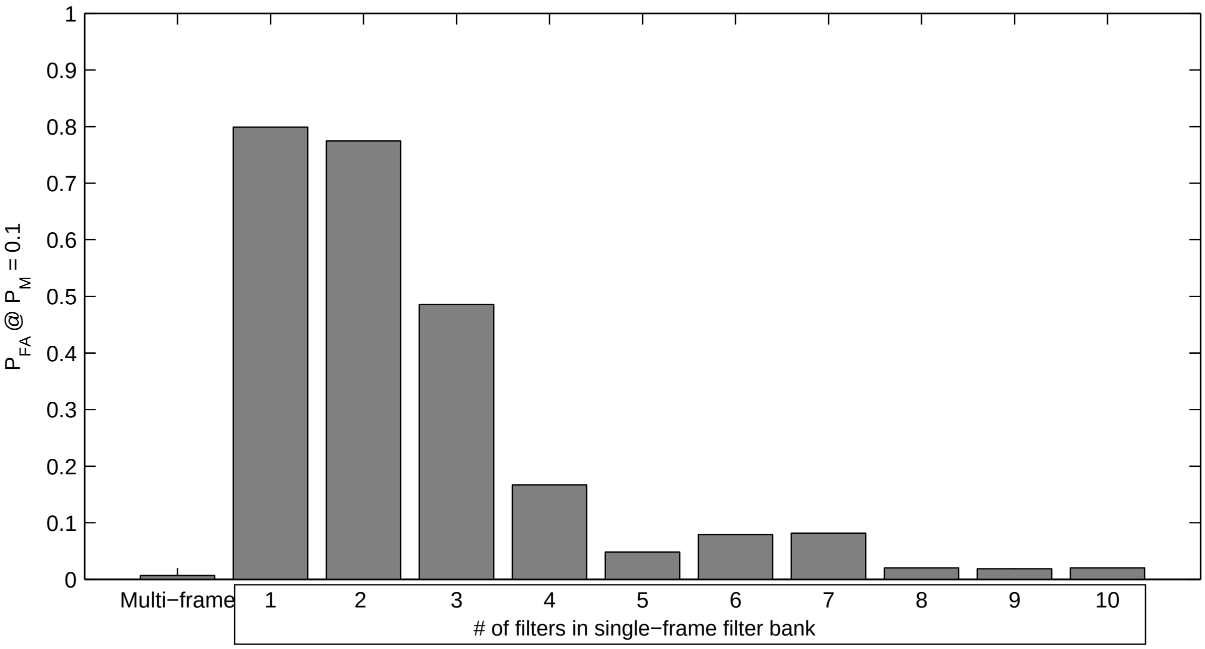


Figure 5. Plot of false alarm rate at a fixed $\mathrm{P_M}$ for different filter bank sizes on real sequence "L1608". The leftmost bar represents multi-frame performance with a single filter, while the others represent single-frame performance.

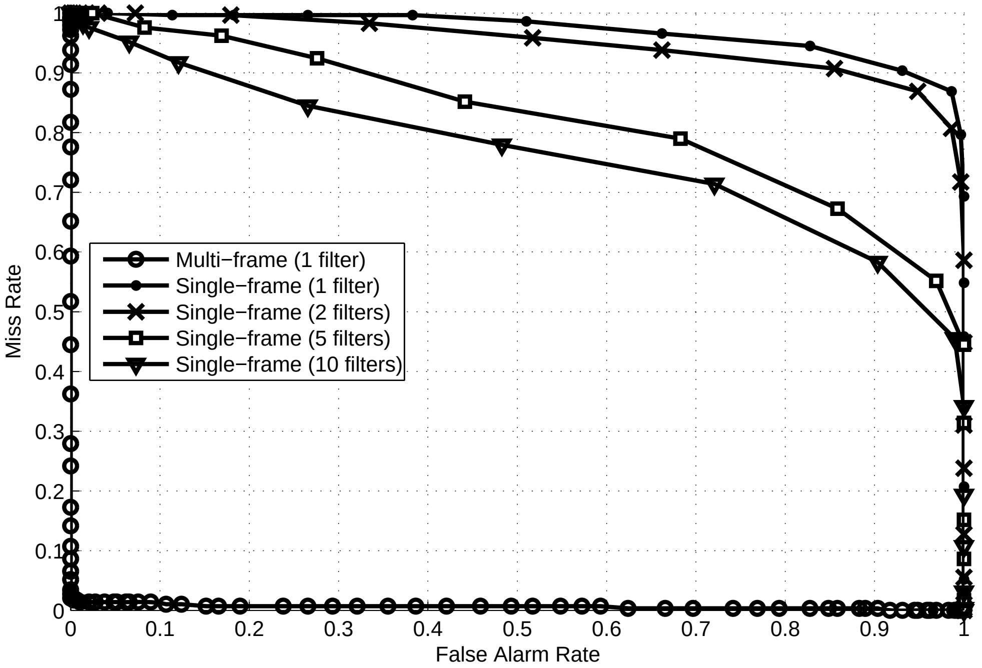


Figure 6. ROC curves for single-frame and multi-frame algorithms on sequence “L1608”. The value $\sigma_{\mathrm{TP}} = 5$ was used to generate the multi-frame curve.

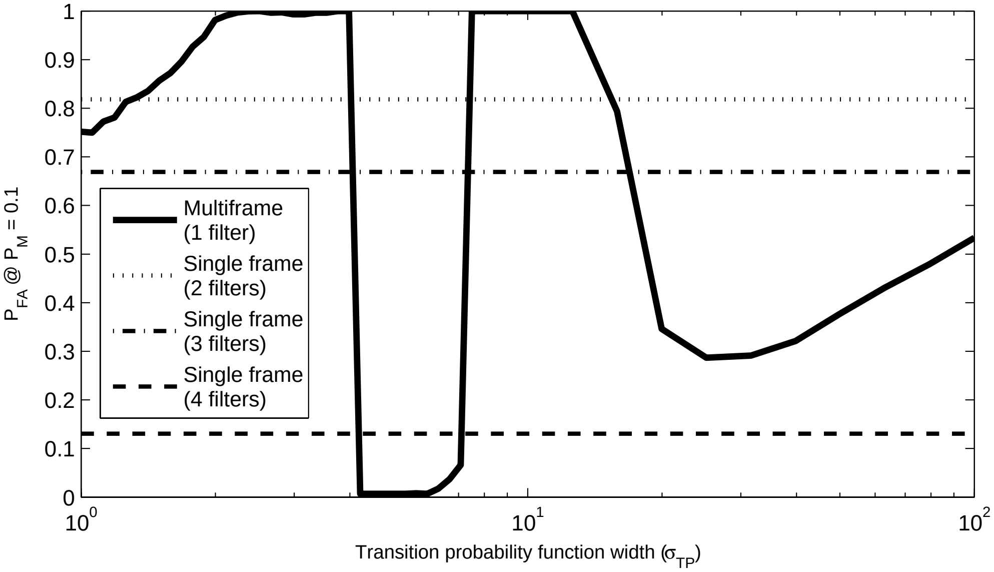


Figure 7. Plot of false alarm rate at $\mathrm{P_M} = 0.1$ for different values of σ_{TP} on sequence “L1608”. Single-frame performance does not depend on σ_{TP} because it is not a parameter in the single-frame algorithm.