

Automatic Target Detection and Tracking in Forward-Looking Infrared Image Sequences Using Morphological Connected Operators

Ulisses Braga-Neto^a

^a*Section of Clinical Cancer Genetics, University of Texas MD Anderson Cancer Center, Houston, TX 77030, U.S.A. and Department of Electrical Engineering, Texas A&M University, College Station, TX 77840, U.S.A.*

Manish Choudhary and John Goutsias^{b,*}

^b*Center for Imaging Science, Clark Hall 308A, The Johns Hopkins University, Baltimore, MD 21218, U.S.A.*

Abstract

We propose a method for automatic target detection and tracking in *forward-looking infrared* (FLIR) image sequences. We use morphological connected operators to extract and track targets of interest and remove undesirable clutter. The design of these operators is based on general size, connectivity and motion criteria, using spatial intraframe and temporal interframe information. In a first step, an image sequence is filtered on a frame-by-frame basis to remove background and residual clutter and to enhance the presence of targets. Detections extracted from the first step are passed to a second step for motion-based analysis. This step exploits the spatiotemporal correlation of the data, stated in terms of a connectivity criterion along the time dimension. The proposed method is suitable for pipelined implementation or time progressive coding/transmission, since only a few frames are considered at a time. Experimental results, obtained with real FLIR image sequences, illustrating a wide variety of target and clutter variability, demonstrate the effectiveness and robustness of the proposed method.

1 Introduction

Automatic target detection and tracking (ATDT) in *forward looking infrared* (FLIR) image sequences is a very important military application. However, ATDT is a difficult task due to the high variability of targets and background clutter and the low spatial resolution of FLIR images.

* Corresponding author. Tel.: +1-410-516-7871; fax +1-410-516-4594; e-mail: goutsias@jhu.edu.

Usually, targets in FLIR images show up either as bright or as dark objects on a background. In many cases, there is not enough contrast between targets and background. Moreover, the background is heavily cluttered. Clutter can be attributed to sensor noise, natural background texture, and to human “artifacts” in the scene (e.g., buildings or other “uninteresting” objects). The background can be highly inhomogeneous and may contribute high-contrast edges that increase the number of false alarms. Manifestations of targets and background clutter in a FLIR image may vary widely, due to changing thermodynamic and atmospheric conditions. Moreover, image illumination may vary spatially and temporally across the image. This results in nonuniform contrast over targets, which may prevent correct detection and extraction of good target contours. In several cases, the targets are very similar to noise and clutter and a human observer may not be able to distinguish them from noise. Usually these, and several other factors, may result in unsatisfactory ATDT performance, in terms of detection and false alarm rates.

In this paper, we propose a scheme for detecting and tracking targets in FLIR image sequences based on a class of morphological operators known as *connected operators* (see Vincent, 1993; Salembier and Serra, 1995; Salembier et al., 1998; Goutsias and Batman, 2000). Our approach results in an effective and robust technique for clutter suppression and normalization, and produces consistent detection performance over a wide range of illumination conditions. Our aim is not to perform accurate target segmentation but to provide reliable target detection with low number of false alarms. In many instances however our scheme provides sufficiently good target segmentation as well.

In this paper, connected operators are designed by means of general size, connectivity and motion criteria, using spatial intraframe and temporal interframe information. These operators are very effective for *image simplification*, in which clutter is suppressed without compromising important target information. This nonlinear approach to image simplification is advantageous over linear approaches that remove frequency content, or other morphological approaches (e.g., by means of median filtering or filtering with morphological openings) that may suppress clutter at the expense of removing valuable target information. Connected operators are excellent in suppressing clutter while selectively preserving valuable contour/shape information.

The proposed ATDT algorithm consists of two steps. The first step involves *intraframe* processing of a given FLIR sequence using simple size and relative position criteria. The second step involves *interframe* processing using a simple motion criterion based on a particular concept of spatiotemporal connectivity. The structure of the second step makes the proposed method suitable for pipelined implementation or time progressive coding and transmission, since it only considers a few frames at a time. The overall method effectively handles the variability issue in FLIR images, since it does not require target modelling. This is in contrast to other pattern-theoretic-based approaches to the same problem (e.g., see Lanterman et al., 1997).

A similar ATDT algorithm for FLIR sequences has been proposed by Rivest and Fortin (1996). The basic conceptual structure of this algorithm is similar to ours. In both schemes,

the first step involves estimating the cluttered background of each frame and subtracting it from the original image by means of morphological top-hat operators. In Rivest and Fortin (1996), the background is estimated by morphological openings. In our scheme, the background is estimated using *opening* and *closing by reconstruction operators*. These are connected operators that provide a better suppression of clutter and better preservation of target shape. Another difference between the two schemes is the way temporal filtering is implemented. The algorithm of Rivest and Fortin (1996) uses an approach based on moving average filtering for target tracking. Instead, we use connectivity criteria, which turn out to be simpler and more powerful for detecting and tracking target motion.

A key tool used in the detection part of both algorithms is thresholding. The algorithm of Rivest and Fortin (1996) performs target detection by simple thresholding, with a threshold value that is kept constant over the entire image sequence. This uniform choice for the value of thresholding may not be appropriate, especially in cases of high clutter and target variability, or in cases of wide changes in environmental conditions. Indeed, it has been reported by Rivest and Fortin (1996) that their algorithm faces difficulties in cases of very low target/background contrast. Our scheme uses an adaptive version of a morphological method for thresholding, known as *double thresholding* (see Soille, 1999). In this case, the level of thresholding changes from frame to frame. Changes are decided by comparing the number of detected targets to an expected number of targets that may appear in a given scene. The main reason why we follow this approach is the intuitive understanding that FLIR images are characterized by wide variations in relative target/background contrast. Finally, we point out that the scheme proposed in this paper is capable of detecting both bright and dark targets, as opposed to the one in Rivest and Fortin (1996), which is based on the assumption that only bright targets are of interest.

This paper is organized as follows. In Section 2, we provide a brief introduction to the notion of connected operators. The main purpose of this section is to discuss the defining property of connected operators and to argue the suitability of these operators for shape simplification, clutter suppression and filtering. In Section 3, we describe the proposed ATDT scheme in detail. In Section 4, we present experimental results obtained by applying our algorithm, as well as the algorithm proposed by Rivest and Fortin (1996), on FLIR image sequence data provided to us by the *U.S. Army Missile Command* (MICOM). Our results clearly demonstrate the effectiveness of the proposed method and its superiority over the technique proposed by Rivest and Fortin (1996). Finally, our conclusions are summarized in Section 5.

2 Connected Operators

Connected operators are filters that do not modify individual pixel values but act at the level of the flat zones of an image. A *flat zone* is a maximally connected region of the domain of definition of an image with a constant gray level value. According to the classical notion of (path) connectivity, a region is connected if each pair of points in the region can be joined by a path whose points are in the region as well. In the discrete case, connectivity reduces to the definition of a local neighborhood describing connections

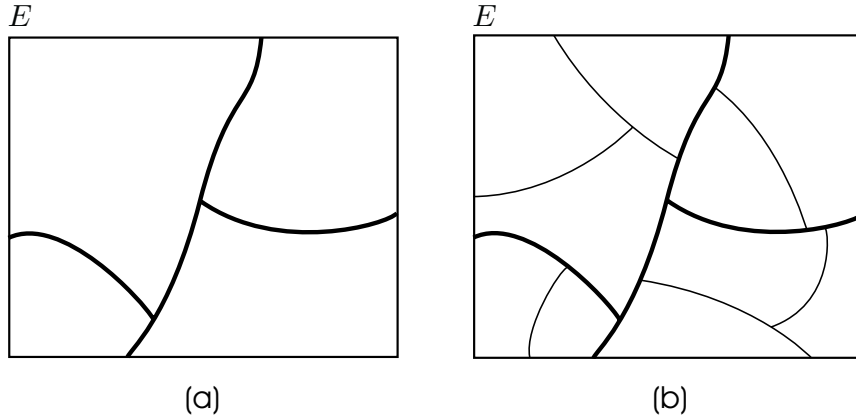


Fig. 1. The partition in (a) is coarser than the one in (b).

among adjacent pixels. There is no restriction on the size of the flat zones, which can be reduced to a single point. The flat zones of a binary image are called *grains* (foreground) and *pores* (background).

The flat zones of an image define a *partition* P of its domain of definition E . The partition P is a collection of connected components $\{A_i\}$, which are disjoint ($A_i \cap A_j = \emptyset, i \neq j$) and whose union forms the entire space ($\bigcup A_i = E$). We can define a partial order between the partitions. A partition $\{A_i\}$ is said to be *coarser* than another partition $\{B_j\}$ if each flat zone of $\{A_i\}$ is contained in some flat zone of $\{B_j\}$; see Fig. 1 for an example. In general, we may not be able to compare any two given partitions.

A connected operator must coarsen the partition generated by the flat zones of an image. Specifically, an operator ψ is said to be a *connected operator* if, for every image f , the partition $P_{\psi(f)}$ of the output image $\psi(f)$ is coarser than the partition P_f of the input image f . We infer that a connected operator can strengthen or weaken boundaries (or even remove them), but it cannot shift boundaries or introduce new ones. Therefore, connected operators enjoy excellent contour preservation properties. An example that illustrates this property is depicted in Fig. 2.

The definition of a connected operator depends on the particular type of connectivity used. Jean Serra has proposed in (Serra, 1988, 1998) the concept of a *connectivity class*, which generalizes the classical notion of connectivity (see also Braga-Neto and Goutsias, 2003). This generalization turns out to be very useful for defining a wide range of connected operators. A (binary) connectivity class \mathcal{C} is a family of binary images defined on E that satisfy the following two properties: (a) The zero image and images consisting of a single pixel belong to \mathcal{C} , and (b) if a collection of images in \mathcal{C} has nonempty intersection, then their union is in \mathcal{C} as well. An example of a useful binary connectivity class is the so-called *dilation-based connectivity class*. If $\mathcal{P}(\mathbb{R}^2)$ is the collection of all subsets of the two-dimensional Euclidean space \mathbb{R}^2 , and if $F \oplus B = \{v + w \mid v \in F, w \in B\}$ is the (translation invariant) dilation of a binary image $F \subset \mathbb{R}^2$ by a structuring element $B \subset \mathbb{R}^2$, then the collection $\mathcal{C}_\delta = \{F \in \mathcal{P}(\mathbb{R}^2) \mid F \oplus B \text{ is connected in the classical sense}\}$ is a connectivity

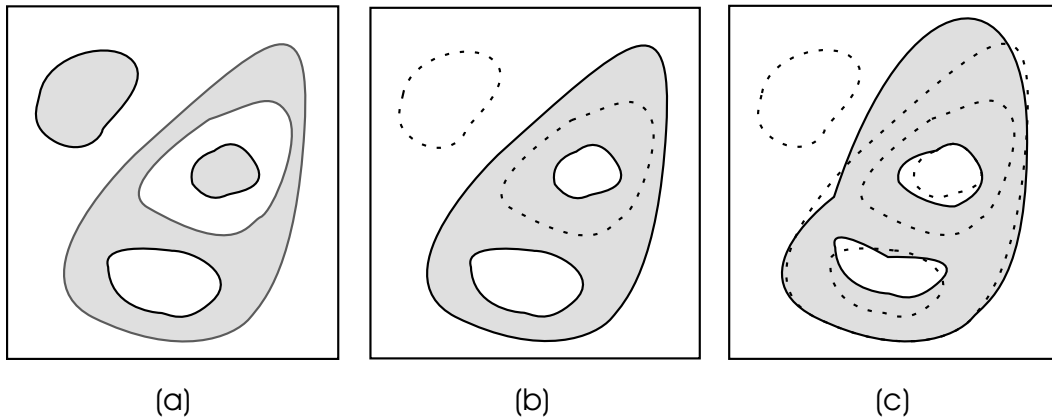


Fig. 2. Example of a binary connected operator: (a) An input image. (b) The output of a connected operator. (c) The output of a non-connected operator. Note that the object in (b) has been obtained from (a) by only removing grains and filling in pores, without modifying object boundaries. Therefore, the partition in (b) is coarser than the partition in (a).

class in $\mathcal{P}(\mathbb{R}^2)$. This implies that a binary image F is connected in the connectivity class \mathcal{C}_δ , if its dilation $F \oplus B$ by a structuring element B is connected in the classical sense. An image F that is not connected in the classical sense may be connected in \mathcal{C}_δ , since the dilation of a disconnected set may be connected. This is illustrated in Fig. 3.

A useful class of binary connected operators are the so-called *reconstruction operators*. These operators extract connected components (grains) of a binary image F , known as the *mask*, that are intersected by another binary image $F_m \subseteq F$, known as the *marker*. In general, reconstruction operators can be expressed by iterating the dilation of marker F_m by an appropriately chosen structuring element C (that contains the pixel $(0, 0)$), making sure that, at each iteration, the dilation is restricted inside the mask F . The basic operator is known as *conditional dilation* of size 1 and is expressed as

$$\delta_C^1(F_m | F) = (F_m \oplus C) \cap F.$$

Iterating this conditional dilation k times yields the conditional dilation of size k , given by

$$\delta_C^k(F_m | F) = \delta_C^1(\cdots \delta_C^1(\delta_C^1(F_m | F) | F) \cdots | F).$$

This operator is useful for extracting all grains \hat{F} of F that are marked by the marker F_m . Indeed, it can be shown that, for an appropriate choice of the structuring element C ,

$$\hat{F} = R_C(F_m | F) = \bigcup_{k \geq 1} \delta_C^k(F_m | F), \quad (1)$$

where the operator $R_C(F_m | F)$ is the reconstruction operator. Due to the fact that the grains in an image F are bounded, a finite number of unions in (1) recovers \hat{F} entirely.

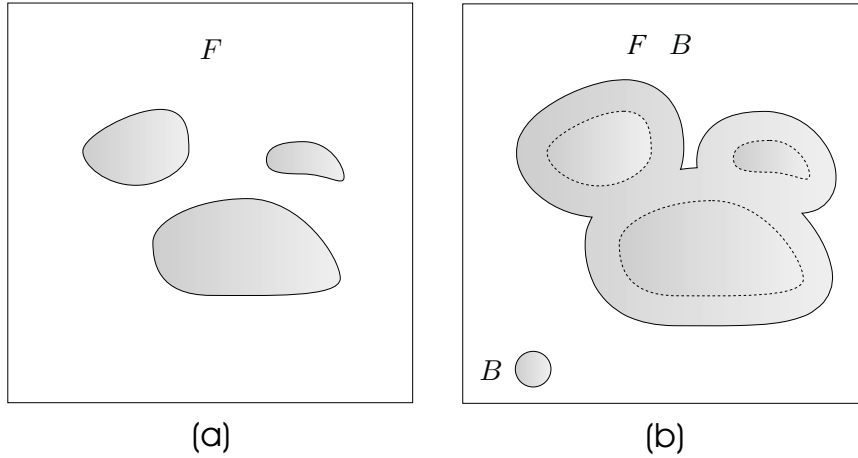


Fig. 3. Example of dilation-based connectivity. (a) The binary image F is not connected. (b) The dilation $F \oplus B$ of F by a disk structuring element B is connected. Therefore, F is considered to be connected in the dilation-based connectivity class \mathcal{C}_δ .

A useful example of a reconstruction operator is obtained by setting $F_m = F \ominus B$, for some structuring element B , where $F \ominus B = \{v \in E \mid B_v \subseteq F\}$ is the (translation invariant) erosion of F by B , with $B_v = \{b + v \mid b \in B\}$ being the translation of B by $v \in E$. The resulting operator, given by

$$\Psi_{\text{openrec}}(F) = R_C(F \ominus B \mid F),$$

is increasing, anti-extensive and idempotent, and is thus an opening (see Heijmans, 1994). This operator reconstructs all grains of F that survive the erosion of F by B , and is known as a (binary) *opening by reconstruction operator*. See Fig. 4(a) for an example.

The *dual* (in terms of set complementation) operator

$$\Psi_{\text{clorec}}(F) = [R_C(F^c \ominus B \mid F^c)]^c$$

is increasing, extensive and idempotent, and is thus a closing (see Heijmans, 1994). This operator reconstructs all pores of F that survive the erosion of F^c by B , and is known as a (binary) *closing by reconstruction operator*.

Since the operator Ψ_{openrec} is anti-extensive, we have that $\Psi_{\text{openrec}}(F) \subseteq F$. Therefore, the set difference

$$\Psi_{\text{openreth}}(F) = F \setminus \Psi_{\text{openrec}}(F)$$

contains all grains of F that are eliminated by the erosion of F by B . This operator is known as a (binary) *opening by reconstruction top-hat operator*. See Fig. 4(b) for an example. Similarly, since Ψ_{clorec} is extensive, we have that $\Psi_{\text{clorec}}(F) \supseteq F$. Therefore, the set difference

$$\Psi_{\text{cloreth}}(F) = \Psi_{\text{clorec}}(F) \setminus F$$

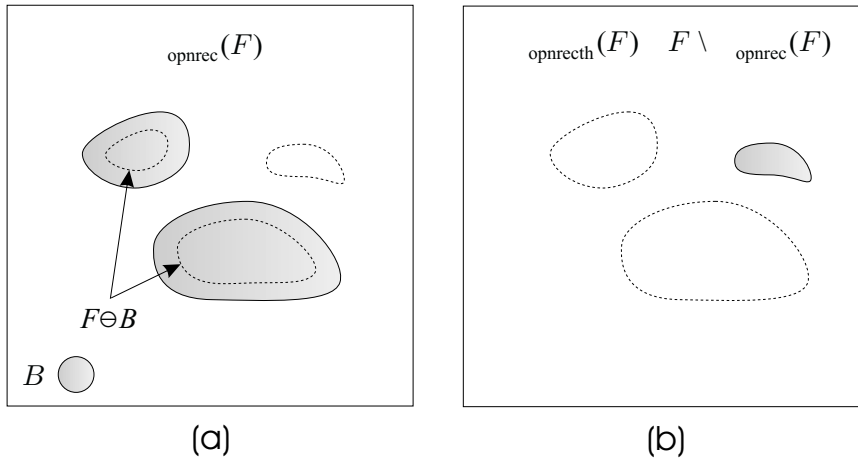


Fig. 4. Example of applying: (a) the opening by reconstruction operator Ψ_{opnrec} on the image F depicted in Fig. 3(a), and (b) the opening by reconstruction top-hat operator Ψ_{opnrecth} on F .

contains all pores of F that are eliminated by the erosion of F^c by B . This operator is known as a (binary) *closing by reconstruction top-hat operator*.

The connected operators discussed so far are binary. To extend those operators to grayscale images, we first define the *cross section* $F(t)$ of a grayscale image f at level t by

$$F(t) = \{(x, y) \in E \mid f(x, y) \geq t\}, \quad t \in \mathbb{R}.$$

The collection $\mathcal{F} = \{F(t) \mid t \in \mathbb{R}, F(t) \neq \emptyset\}$ of all nonempty cross sections of a grayscale image f constitutes the *threshold decomposition* of f . This provides a unique and reversible decomposition of an image f into a collection \mathcal{F} of binary images. The value of f at pixel (x, y) can be obtained by means of

$$f(x, y) = \bigvee \{t \in \mathbb{R} \mid (x, y) \in F(t)\},$$

where \bigvee denotes *supremum* (maximum).

Threshold decomposition provides a way to construct grayscale morphological operators by means of binary ones. Given an *increasing* binary operator Ψ , a grayscale operator ψ can be constructed by applying Ψ on the cross sections $F(t)$, $t \in \mathbb{R}$, of a grayscale image f and by setting

$$\psi(f)(x, y) = \bigvee \{t \in \mathbb{R} \mid (x, y) \in \Psi(F(t))\}, \quad (x, y) \in E.$$

The operator ψ is called the *flat grayscale operator* generated by Ψ . Note that the cross section of $\psi(f)$ at level t equals $\Psi(F(t))$, provided that Ψ is \downarrow -continuous (see Heijmans, 1994, p. 48, for the definition of an \downarrow -continuous set operator), or the grayscale values are finite.

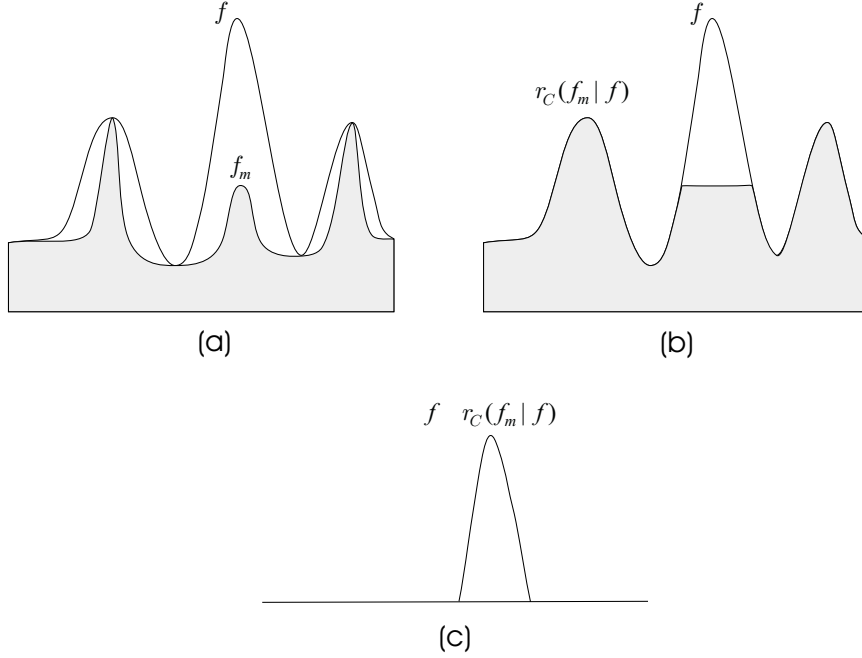


Fig. 5. (a) A grayscale signal f and a marker f_m that identifies the portion of f that needs to be preserved. (b) The operator $r_C(f_m | f)$ reconstructs that portion of f that is marked by f_m . (c) The difference $f - r_C(f_m | f)$ contains the peak of f that is not marked by f_m .

We are now interested in an operator that extracts a pre-selected number of peaks in an image f , while leaving the rest of the image intact. Let $f_m \leq f$ be a marker image, which identifies the portion of f that needs to be preserved; see Fig. 5(a). This can be done in three steps. First, we calculate the threshold decompositions $\mathcal{F} = \{F(t) \mid t \in \mathbb{R}, F(t) \neq \emptyset\}$ and $\mathcal{F}_m = \{F_m(t) \mid t \in \mathbb{R}, F_m(t) \neq \emptyset\}$. Then, we reconstruct, for each t , all grains of $F(t)$ that are marked by $F_m(t)$ using the binary reconstruction operator $R_C(F_m(t) \mid F(t))$ (since $f_m \leq f$, we have $F_m(t) \subseteq F(t)$, for each t). Finally, we set

$$r_C(f_m | f)(x, y) = \bigvee \{t \in \mathbb{R} \mid (x, y) \in R_C(F_m(t) \mid F(t))\}.$$

The resulting flat grayscale operator $r_C(f_m | f)$ is known as a (grayscale) *reconstruction operator* and is a connected operator (see Serra and Salembier, 1993; Braga-Neto, 2001). Clearly, $r_C(f_m | f)$ reconstructs that portion of image (mask) f that is marked by (marker) f_m ; see Fig. 5(b). This is done by means of individually reconstructing each binary cross section. Keep in mind however that, in practice, the grayscale reconstruction operator is implemented by a different and much faster technique (Vincent, 1993).

Choosing an appropriate marker f_m is possibly the most important aspect of using reconstruction operators in target detection problems. “Hot” targets express themselves as grayscale peaks in the intensity profile of an image f . The erosion $F(t) \ominus B$ of the cross section $F(t)$ of image f by a structuring element B that contains pixel $(0, 0)$ and that is slightly larger than the target grains in $F(t)$ will remove all such grains and will provide a marker for the clutter grains. Using the flat grayscale erosion

$$(f \ominus B)(x, y) = \bigvee \{t \in \mathbb{R} \mid (x, y) \in F(t) \ominus B\} = \bigwedge_{(v,w) \in B} f(x+v, y+w)$$

as a marker, where \wedge denotes *infimum* (minimum) (note that $f \ominus B \leq f$), and image f as a mask, we can reconstruct the clutter. The operator

$$\psi_{\text{opnrec}}(f) = r_C(f \ominus B \mid f) \quad (2)$$

removes peaks from image f , whereas it has minimal effect on the rest of f . This operator is known as a (grayscale) *opening by reconstruction operator*. It is an increasing, anti-extensive and idempotent operator, and thus it is an opening.

“Cold” targets appear as dips in the intensity profile. We can use the *dual* operator

$$\psi_{\text{clorec}}(f) = [r_C(f^* \ominus B \mid f^*)]^* \quad (3)$$

to remove the dips from image f while leaving the rest of f intact, where $f^* = -f$, for continuous-valued images, or $f^* = M - f$, for discrete-valued images, with M being the maximum gray level value. This operator is increasing, extensive, and idempotent, and thus it is a closing. It is known as the (grayscale) *closing by reconstruction operator*.

Since the operator ψ_{opnrec} is anti-extensive, we have that $\psi_{\text{opnrec}}(f) \leq f$. Therefore, the difference

$$\psi_{\text{opnreth}}(f) = f - \psi_{\text{opnrec}}(f)$$

is nonnegative and contains all peaks of f that are eliminated by the opening by reconstruction operator in (2). This operator is known as a (grayscale) *opening by reconstruction top-hat operator*. See Fig. 5(c) for an example. Similarly, since ψ_{clorec} is extensive, we have that $\psi_{\text{clorec}}(f) \geq f$. Therefore, the difference

$$\psi_{\text{cloreth}}(f) = \psi_{\text{clorec}}(f) - f$$

is nonnegative and contains all dips of f that are eliminated by the closing by reconstruction operator in (3). This operator is known as a (grayscale) *closing by reconstruction top-hat operator*.

3 A Reconstruction-Based ATDT Algorithm

We now describe the two main steps of our FLIR ATDT algorithm. No model library of specific targets is required. We only make three basic and purely geometrical assumptions regarding the targets of interest. These assumptions correspond to the following criteria:

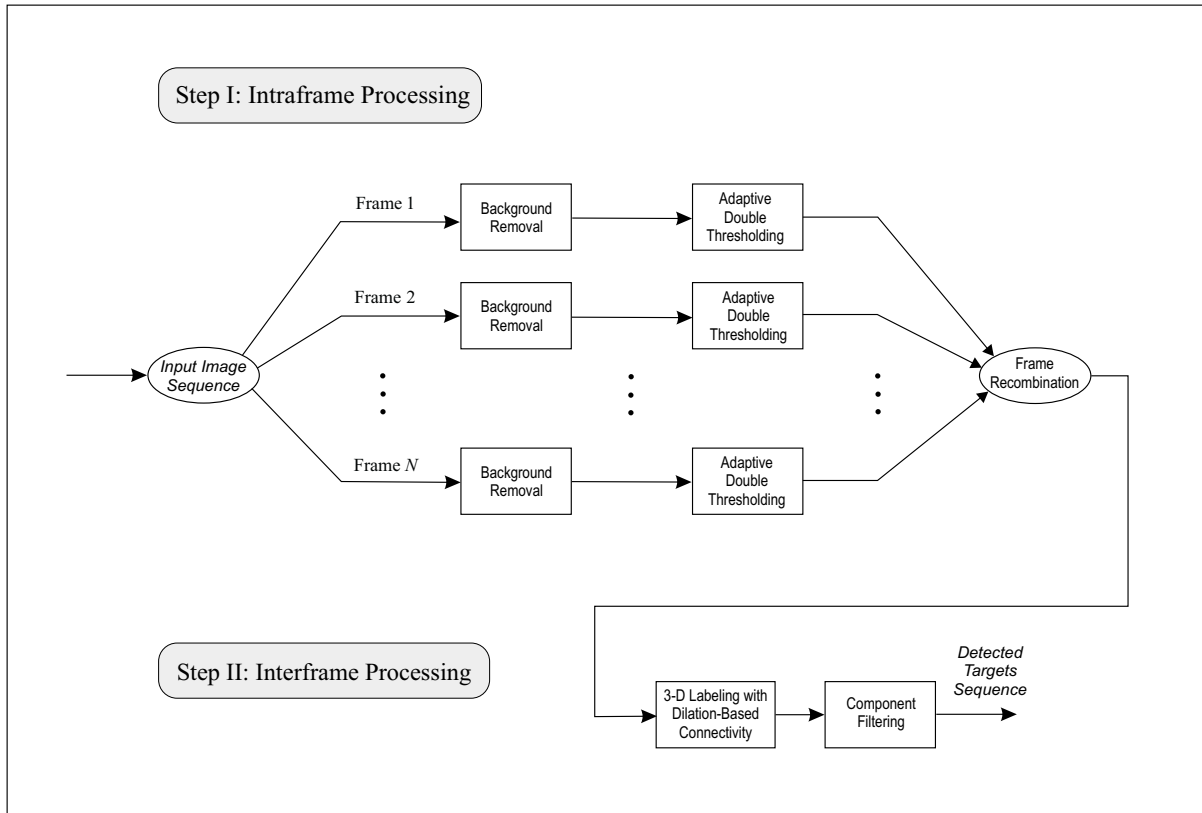


Fig. 6. Block-diagram of the FLIR ATDT algorithm.

- **Size:** The targets of interest have a maximum specified apparent size.
- **Relative Position:** The targets of interest are situated away from the boundary of the *field of view* (FOV).
- **Motion:** The targets of interest have limited relative motion with respect to the FLIR sensor.

These criteria reflect natural assumptions about the targets of interest. The *size* criterion means that very large features, such as roads, buildings, etc., are likely to be clutter. This assumption helps greatly in simplifying the background. We have been confronted with situations where targets of interest do not satisfy this criterion (e.g., when the camera is close to a target, in which case the size of the target is quite large). However, we get unhindered detection of these targets owing to the fact that spatial illumination may not be uniform over a target. In this case, large targets contribute much smaller signatures (peaks or dips), which satisfy our *size* criterion. The *relative position* criterion states that targets of interest situated at the periphery of the sensor’s FOV cannot be reliably detected (in addition, clutter is likely to extend beyond the FOV). The *motion* criterion assumes that the relative speed of the targets of interest with respect to the sensor’s FOV is limited, so that they can be reliably detected and tracked (this assumption is true for the vast majority of available FLIR data).

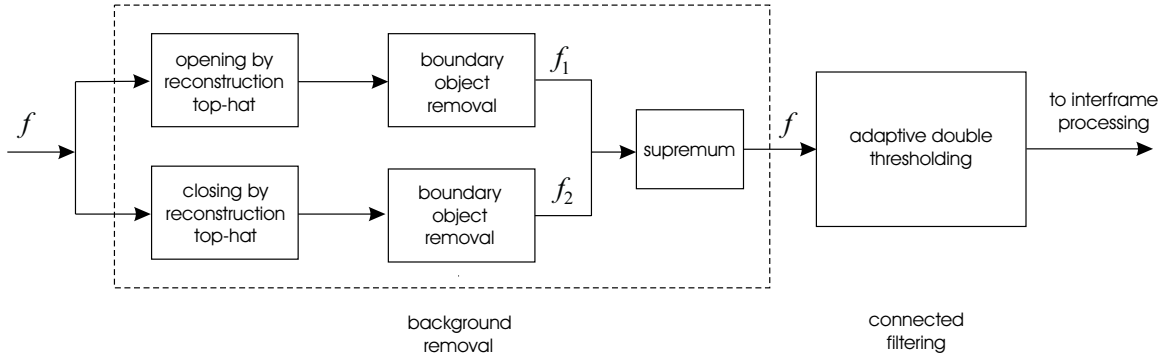


Fig. 7. Details of the intraframe processing step.

A block diagram of our FLIR ATDT approach is depicted in Fig. 6. Our algorithm processes video sequences in two steps. During the first step (Step I – Intraframe Processing), the individual frames of an input image sequence are processed independently. The purpose of this step is to detect targets in individual image frames based on contrast. “Hot” targets express themselves as peaks, whereas “cold” targets express themselves as dips in the image profile. The intraframe processing step detects peaks and dips in order to get all possible candidate profiles and processes them in order to narrow down the ones that correspond to targets of interest. During the second step (Step II – Interframe Processing), false alarms are removed by exploiting spatiotemporal correlations in the data, stated in terms of a modified connectivity criterion. In the rest of this section, we discuss both steps in detail.

3.1 Intraframe Processing

Figure 7 provides details of the intraframe processing step. This step consists of two substeps: background removal and connected filtering.

3.1.1 Background Removal

This substep primarily uses reconstruction top-hat operators to reduce background clutter and enhance the presence of targets. As we mentioned earlier, the choice of an appropriate marker is the most important aspect of using reconstruction operators for target detection. In this regard, a reasonably good estimate of target geometry is critical for satisfactory performance.

Our algorithm uses two opening and closing by reconstruction top-hat operators applied on each frame in parallel. The same structuring element B is used in both reconstruction operators. Opening by reconstruction $\psi_{\text{opnrec}}(f) = r_C(f \ominus B | f)$ is applied on an input frame f to effectively reclaim all portions of f below the marker $f \ominus B$. Then, the target peaks are extracted by means of the opening by reconstruction top-hat operator $\psi_{\text{opnrecth}}(f) = f - r_C(f \ominus B | f)$. This operator extracts peaks that correspond to “hot” targets. A dual closing by reconstruction top-hat operator $\psi_{\text{clorecth}}(f) = [r_C(f^* \ominus B | f^*)]^* - f$ is also applied on the frame. This operator extracts dips that correspond to “cold” targets.

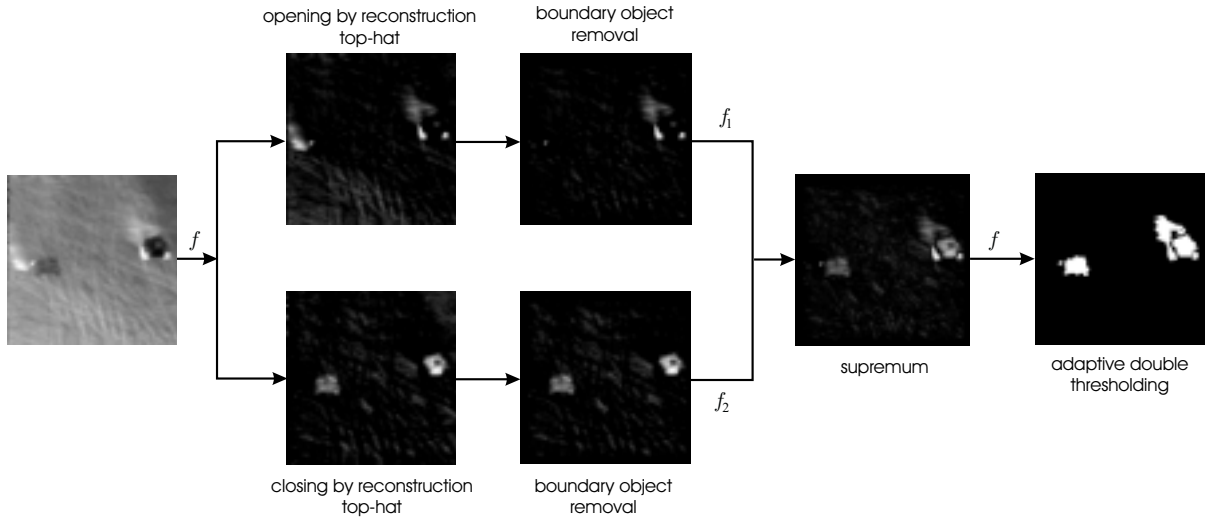


Fig. 8. Illustration of results obtained by the intraframe processing step depicted in Fig. 7.

The overall process removes most background clutter, while preserving the shape of “hot” and “cold” targets.

Recall that $\psi_{\text{opnrec}}(f) \leq f$ and $\psi_{\text{clorec}}(f) \geq f$, which implies that $\psi_{\text{opnrecth}}(f) \geq 0$ and $\psi_{\text{clorecth}}(f) \geq 0$. Therefore, the opening and closing by reconstruction top-hat operators produce images, which contain both “hot” and “cold” targets that appear as bright objects contrasted on a dark background; see Fig. 8. These objects are “smaller” than the structuring element B used, in agreement with our *size* criterion. Note that the reconstruction operators, being connected operators, recover most of background without shifting boundaries. Therefore, the overall background removal, implemented by reconstruction top-hat operators, is very effective.

Subsequently, all targets that touch a frame of width w around the image boundary are discarded. This is in agreement with our *relative position* criterion, and is accomplished by means of subtracting from the top-hat image the grayscale reconstruction result that recovers targets that touch the boundary frame. The marker f^{border} for the grayscale reconstruction operator is taken to be zero everywhere except along the boundary frame. This produces two images f_1 and f_2 , given by

$$f_1 = \psi_{\text{opnrecth}}(f) - r_C(f^{\text{border}} \mid \psi_{\text{opnrecth}}(f)),$$

$$f_2 = \psi_{\text{clorecth}}(f) - r_C(f^{\text{border}} \mid \psi_{\text{clorecth}}(f)),$$

which contain only targets that are far enough from the boundary; see Fig. 8.

Finally, f_1 and f_2 are combined by taking the pixel-wise supremum. This produces a new image f' , given by $f'(x, y) = (f_1 \vee f_2)(x, y)$, which contains all target candidates (“hot” and “cold”), with each candidate appearing as a bright peak; see Fig. 8. Since illumination varies spatially, these peaks may not have equal strength. We use “peak strength” as a

criterion to distinguish between targets and false alarms, based on the premise that, most likely, a stronger peak corresponds to a valid target. This leads to the next substep of adaptive double thresholding.

3.1.2 Adaptive Double Thresholding

This substep produces a binary image of targets by thresholding the image f' obtained from the previous substep and by removing residual clutter; see Fig. 8. We assume that small peaks are due to background clutter that has escaped the previous substep, and use thresholding to remove these (unwanted) peaks. A simple thresholding operator $T_{[t_i, t_j]}$, known as *level slicing*, sets all pixel (x, y) of an input image f whose values are in a pre-specified range $[t_i, t_j]$ to 1 and all other values to 0. In this case

$$T_{[t_i, t_j]}(f)(x, y) = \begin{cases} 1, & \text{if } t_i \leq f(x, y) \leq t_j \\ 0, & \text{otherwise} \end{cases}.$$

However, the use of simple thresholding suffers from two major drawbacks: (a) a large difference in slicing values results in an image contaminated by clutter, and (b) a small difference results in an image that may contain targets of interest, but split into disconnected regions, especially in cases of nonuniform illumination. Simple thresholding is very sensitive to the chosen slicing values and a slight change in those values may even completely eliminate the targets of interest. For these reasons, it is very difficult to find unique slicing values for all frames in a sequence that result in satisfactory ATDT performance.

To overcome these difficulties, we have decided to use *double thresholding* (see Soille, 1999). This type of thresholding is based on morphological reconstruction and is quite robust. The associated parameters are found adaptively for each frame. The scheme provides superb ATDT performance over a wide range of image sequences.

The double thresholding operator slices the gray levels of an input image by using two ranges of grayscale values, one included in the other. Let the two ranges be $[t_1, t_4]$ and $[t_2, t_3]$, such that $[t_1 \leq t_2 \leq t_3 \leq t_4]$. Observe that $T_{[t_2, t_3]}(f) \subseteq T_{[t_1, t_4]}(f)$. We can therefore use $T_{[t_2, t_3]}(f)$ as a marker and $T_{[t_1, t_4]}(f)$ as a mask for binary image reconstruction. This leads to the double threshold operator, given by

$$DT_{[t_1 \leq t_2 \leq t_3 \leq t_4]}(f) = R_C(T_{[t_2, t_3]}(f) | T_{[t_1, t_4]}(f)).$$

If the slicing values are chosen appropriately, then the *wide* level slicing result $T_{[t_1, t_4]}(f)$ will contain both targets and clutter, whereas the *narrow* level slicing result $T_{[t_2, t_3]}(f)$ will contain only information pertaining to the targets. In this case, the reconstruction $R_C(T_{[t_2, t_3]}(f) | T_{[t_1, t_4]}(f))$ will remove unwanted clutter and reconstruct the targets. We set both t_3 and t_4 equal to the maximum value of image f . The double threshold operator is illustrated in Fig. 9.

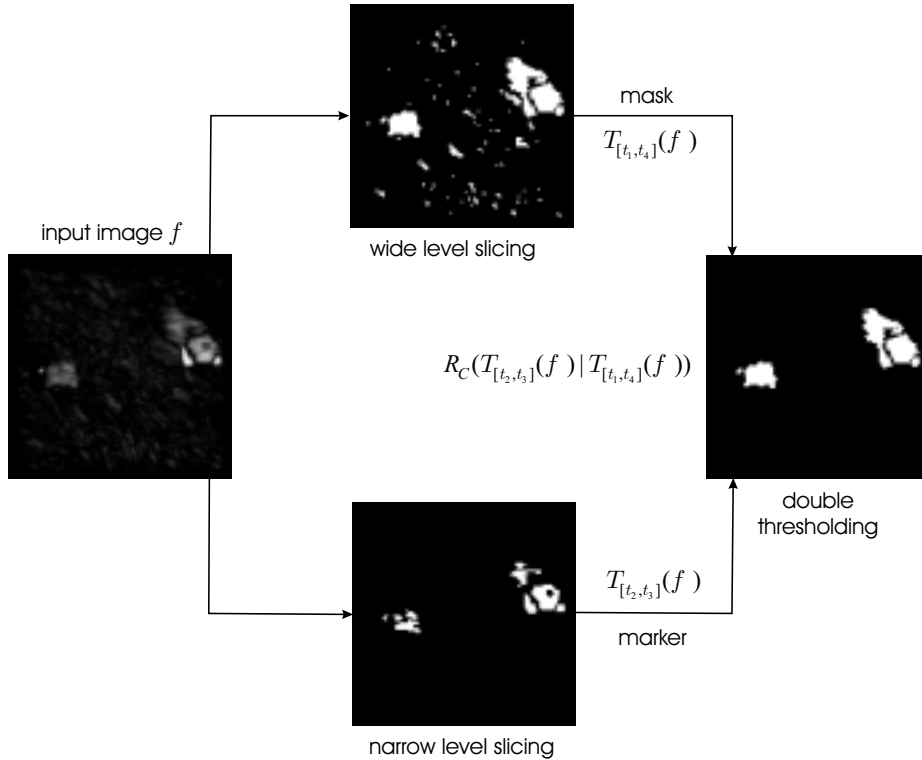


Fig. 9. Illustration of double thresholding. The gray levels of the input image f' are sliced twice: for a narrow and wide range of grayscale values. The images obtained by wide and narrow level slicing are used as mask and marker, respectively, in a binary reconstruction step that extracts targets of interest and removes clutter.

On the other hand, double thresholding avoids, to a large extent, the previously mentioned difficulties associated with simple thresholding. The use of double thresholding produces images with suppressed clutter, thus significantly reducing the number of missed targets and false alarms. The use of reconstruction greatly helps suppressing variations in illumination within targets, thus making it possible to recover the target geometry. A big advantage of double thresholding is its robustness to changes in slicing values. However, the overall ATDT performance still depends on the choice for those values. Since there is frame-to-frame variation in illumination and clutter, a single set of slicing values cannot guarantee satisfactory performance. We achieve excellent performance by *adaptively* modifying the slicing values as we explain next.

Let M be the maximum grayscale value in a given frame and let T_0 be the number of targets of interest that we expect to be present in a given FLIR sequence. The adaptive double thresholding scheme we employ in this paper consists of the following steps:

STEP 1: Set the wide level slicing values at $t_1 = cM$ and $t_4 = M$, for some $c \in (0, 1)$.

STEP 2: Set the narrow level slicing values at $t_2 = dM$ and $t_3 = M$, with $d = c + \epsilon$, for some small $\epsilon > 0$.

STEP 3: Apply double thresholding to image f' , using the previous level slicing values.

STEP 4: Calculate the resulting number of detections D (which includes both true targets and false alarms).

STEP 5: If $D > T$, for some integer $T \geq T_0$, and $d \leq 1 - \epsilon$, set $d = d + \epsilon$, $t_2 = dM$, and go to Step 3.

STEP 6: If $D \leq T$ or $d > 1 - \epsilon$, STOP.

Clearly, given the wide level slicing value t_1 , the previous steps calculate the proper narrow level slicing value t_2 such that double thresholding produces an image that contains no more than T targets. It will become clear in Section 4 that this scheme produces excellent ATDT results, in terms of reducing the number of missed targets and false alarms, even in sequences with low contrast and heavily cluttered frames. The choices for the underlying parameters will also be discussed in Section 4.

3.2 Interframe Processing

After recombining the binary detection results of the previous intraframe processing step into one sequence, the majority of residual clutter and false alarms that survive are removed during a subsequent interframe processing step (Step II). This step exploits spatiotemporal correlation in the data, stated in terms of a dilation-based connectivity criterion along the time direction. It consists of two substeps: 3-D labelling with dilation-based connectivity and component filtering.

3.2.1 3-D Labelling with Dilation-Based Connectivity

The purpose of this substep is to label the binary detections of the intraframe step, so that detections associated with the same target carry the same label. To do so, detections are clustered into distinct groups, with each group containing temporally connected grains. Temporal connectivity is characterized by means of a dilation-based connectivity criterion (see Section 2), which determines that two grains in two consecutive frames are temporally connected if the union of their dilations with a disk structuring element of radius r is connected. This implies that the grains, when superimposed on a single frame, are separated by no more than $2r$, where r is the radius of the disk structuring element used in the dilation; see Fig. 10 for an example. The dilation acts on each frame separately and compensates for target displacement between two consecutive frames. Using dilation-based connectivity, we label grains that are temporally connected across two consecutive frames with the same label. The labelling scheme consists of the following steps:

STEP 1: Let $i = 1$ be the starting frame, and assign unique labels, say $1, 2, \dots, k_1$ to each grain of that frame.

STEP 2: Set $i = i + 1$.

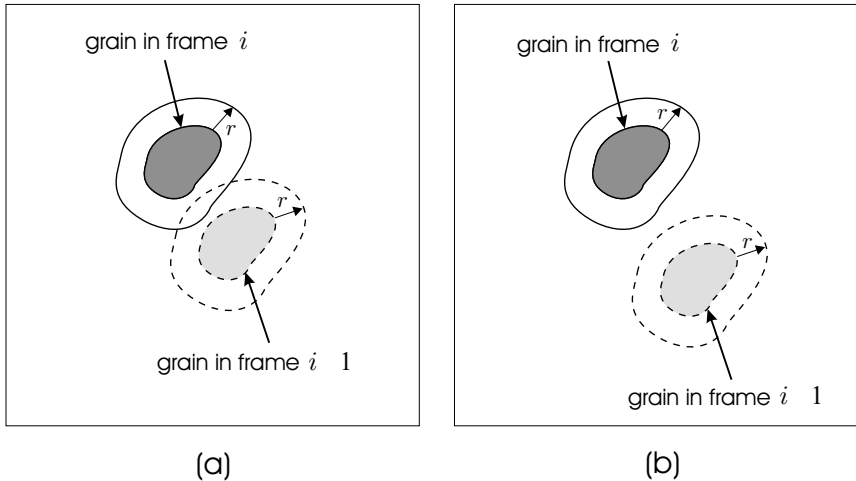


Fig. 10. (a) Two grains that belong to the same target (since their dilations with a disk structuring element of radius r are connected along the time direction). (b) Two grains that belong to two different targets. Note that the grains in (b) may be thought to belong to the same target for a sufficiently large radius of the structuring element.

STEP 3: For each grain $F^{(i)}$ in frame i , find a grain $F^{(i-1)}$ in frame $i - 1$ so that $F^{(i-1)}$ is temporally connected to $F^{(i)}$, and assign to $F^{(i)}$ the same label as the one assigned to $F^{(i-1)}$.

STEP 4: Label the grains in frame i that are not temporally connected to any grains in frame $i - 1$ by $k_{i-1} + 1, k_{i-1} + 2, \dots, k_i$.

STEP 5: Go to Step 2 and repeat until all frames are properly labelled.

Note that detections carrying the same label across the sequence are associated with the same target, which evolves as a function of time. We assume that the velocity of target motion is within a limited range (according to our *motion* criterion). In this case, the radius r of the disk structuring element used in the dilation-based connectivity can be determined from knowledge of such velocity. We use an isotropic structuring element, since the direction of target motion is not known a priori.

3.2.2 Component Filtering

The purpose of this substep is to remove grains that are not consistently detected in the sequence. These grains are thought to be missed targets, false alarms, or targets that are moving too fast to satisfy our *motion* criterion. We assume that a valid target should be detected in at least m consecutive frames. Once the FLIR sequence has been labelled by the previous substep, grains with the same label that do not appear in m consecutive frames are discarded.

We remark that the previous ATDT scheme is very well suited to pipeline processing, or time progressive coding/transmission, since only a few frames are needed at a time. Initially,

m frames are read from the input sequence to form an *active* pile that is processed by Step I on a frame-by-frame basis. The detected grains in the active pile are labelled by the “3-D labelling with dilation-based connectivity” substep of Step II, and the top frame is checked for grains that do not satisfy the criterion imposed by the “component filtering” substep. The grains that pass the test are marked in all m frames. The grains that do not pass the test are removed from the first frame, which is then coded and transmitted. This frame is removed from the pile, and the next unprocessed frame in the sequence is passed through Step I and appended at the bottom of the pile to form a new active pile. The process is repeated, until all frames in the sequence are processed, noting that the marked grains should not be retested nor removed. When there are fewer than m frames left, the entire active pile can be processed as a whole.

4 Experimental Results

We have applied our algorithm on real FLIR video data provided to us by MICOM. These data have been obtained by a FLIR sensor mounted on an airborne platform. ATDT is very challenging in this case. Some data are very cluttered, with targets hiding behind other objects. Moreover, the data contains several targets that are difficult to see in a single frame but become apparent due to motion.

The proposed algorithm is implemented on a Windows 2000/Intel platform. The data consists of several FLIR image sequences. The length of each sequence varies from data set to data set. A frame in an image sequence is stored as an 128×128 pixel 8-bit image with an 8-byte header. The images are read by skipping the first 8 bytes and stored in a standard .tif format.

In addition to the images, MICOM provides ground truth information. Each ground truth file lists all true targets that appear in each frame. The ground truth file specifies the (x, y) coordinates of each target. It is not however clear whether or not these coordinates refer to the centroid of the target or some other feature. The available ground truth information is used to locate the actual targets in a scene.

To test the performance of our algorithm, we first center a 5×5 pixel rectangle at each pair of ground truth target coordinates. The grains detected by our algorithm are then compared with the ground truth targets, which appear as 5×5 rectangular blocks. We use a very simple method for deciding whether detected grains are true detections or false alarms. If a detected grain intersects a rectangular block, then it is classified as true detection; otherwise, it is classified as a false alarm. When more than one detected grains intersect a rectangular block, we assume that all such grains correspond to a single target. The number of false alarms is now the difference between all detected grains minus the grains that have been classified as true detections.

The proposed algorithm requires specification of *nine* parameters. The first two parameters are the structuring elements B and C , used in the opening and closing by reconstruction top-hat operators of the intraframe processing step. The choice of B depends on the geometry of the targets of interest and is determined empirically. We set B to be a

25×25 pixel SQUARE structuring element, since the targets of interest are mostly square or rectangularly shaped. We have chosen the size of B according to the maximum target size observed in the FLIR sequences under consideration. We have observed that the choice of B has a rather moderate effect on ATDT performance. The structuring element C is taken to be the RHOMBUS structuring element $\{(-1, 0), (0, 0), (1, 0), (0, -1), (0, 1)\}$. This choice is dictated by the background connectivity, which is taken to be the 4-connectivity.

The third parameter is the width w of the boundary frame used for boundary object removal in the intraframe processing step. This parameter is chosen by keeping in mind that higher values may result in more targets of interest to be eliminated. We have experimentally determined that setting $w = 8$ pixels is sufficient for the data used in this paper.

The fourth parameter is the threshold value t_1 associated with the wide level slicing step of double thresholding. Since the thermal signatures of a target can vary, the relative strength of target peaks may differ significantly (some peaks may be weaker than other peaks of the same target). The threshold value t_1 is chosen to preserve these weak target peaks while removing very weak peaks that correspond to clutter. This is ensured by assigning a very low value to t_1 , typically in the closed interval $[M/10, M/5]$, where M is the maximum gray level value. We take $t_1 = M/10$ and, since $t_1 = cM$, we set $c = 0.1$. We have observed that the choice for t_1 (or c) has a moderate effect on ATDT performance.

The choice of the initial threshold value t_2 associated with the narrow level slicing step of double thresholding is crucial. Large values of t_2 are responsible for removing most of background clutter and for preserving only strong enough peaks. The value of t_2 lies in the range (t_1, M) . As t_2 tends towards t_1 , the number of actual detections increases, at the cost of increasing the number of false alarms. However, a high value of t_2 may result in missing some targets completely. Recall that $t_2 = dM$, where $d = c + \epsilon$, for some small $\epsilon > 0$. A small value of ϵ is required in order to obtain sufficiently smooth adaptation in thresholding. If ϵ is large, then adaptive thresholding may suddenly eliminate some targets of interest. On the other hand, if ϵ is very small, then adaptive thresholding will become time consuming. We have determined that $\epsilon = 0.05$ is sufficiently good for the data under consideration.

As we have discussed earlier, our algorithm automatically determines the best value of t_2 under the constraint that the number of detections is no more than T , for some integer $T \geq T_0$, where T_0 is the number of targets of interest that we expect to be present in a FLIR sequence. The value of T_0 can be either determined experimentally or by making a reasonable estimate using some a priori knowledge. We have determined the appropriate value of T_0 for each FLIR sequence from the available ground truth data, and we have found that the value $T = 4T_0$ works well for the data used in this paper.

The remaining two parameters are associated with the interframe processing step. The first parameter is the radius r of the disk structuring element in the dilation that determines the spatiotemporal connectivity. This radius depends on relative target motion. The

dilation connects moving targets across frames. We assume that targets of interest have limited relative motion with respect to the FLIR sensor. In this case, r is chosen to connect only those moving targets that satisfy this assumption. We have empirically determined that an appropriate value is $r = 4$. The last parameter associated with the interframe processing step is the number m of consecutive frames in which a valid target should at least be detected. We assume that a spatiotemporal connected component that does not satisfy this criterion consists of residual clutter that appears in just a few consecutive frames or of targets that are moving fast with respect to the FLIR sensor. We have chosen $m = 10$. This value works very well for the available data. We have experimentally observed that the values of r and m have little influence on ATDT performance, unless these values change dramatically.

We have computed ROC curves to measure ATDT performance. An ROC curve plots the probability of correct detection P_d as a function of the number F_a of false alarms per frame in a video sequence, given by:

$$P_d = \frac{\text{number of detections in the sequence}}{\text{number of actual targets in the sequence}},$$

and

$$F_a = \frac{\text{number of false detections in the sequence}}{\text{number of frames in the sequence}}.$$

We have measured the performance of our algorithm by setting $T = T_0, T_0 + 1, \dots$, while keeping all other parameters fixed, and we have compared it to the performance of the Rivest & Fortin algorithm (Rivest and Fortin, 1996) by varying the detection threshold. We refer to our algorithm as the BCG algorithm and to the Rivest & Fortin algorithm as the RF algorithm.

In order to implement and apply the RF algorithm on our data, and fairly compare it with the BCG algorithm, we have removed the overflow protection used by the RF algorithm after the temporal processing step, which clips any gray level value above 255 to 255. This is required in certain cases, since overflow protection clips many false alarms, despite the fact that these false alarms may have lower contrast than the targets. If the background is not dim enough, this problem is encountered very often and gives rise to a large number of false alarms for the RF algorithm.

Figure 11 depicts typical ROC curves for both algorithms and for three representative FLIR video sequences. Figure 11(a) depicts the ATDT performance obtained for a typical case of low-contrast frames and target illumination that is similar to the background. The BCG algorithm detects targets of interest more accurately than the RF algorithm, for the same number of false alarms per frame. Moreover, the ROC curve of the BCG algorithm sharply approaches one, which indicates almost perfect detection performance with few false alarms per frame. Figure 11(b) depicts the ATDT performance obtained for a typical case of high-contrast frames and low cluttered background. Similarly to the results in Fig. 11(a), the BCG algorithm detects targets of interest more accurately than the RF

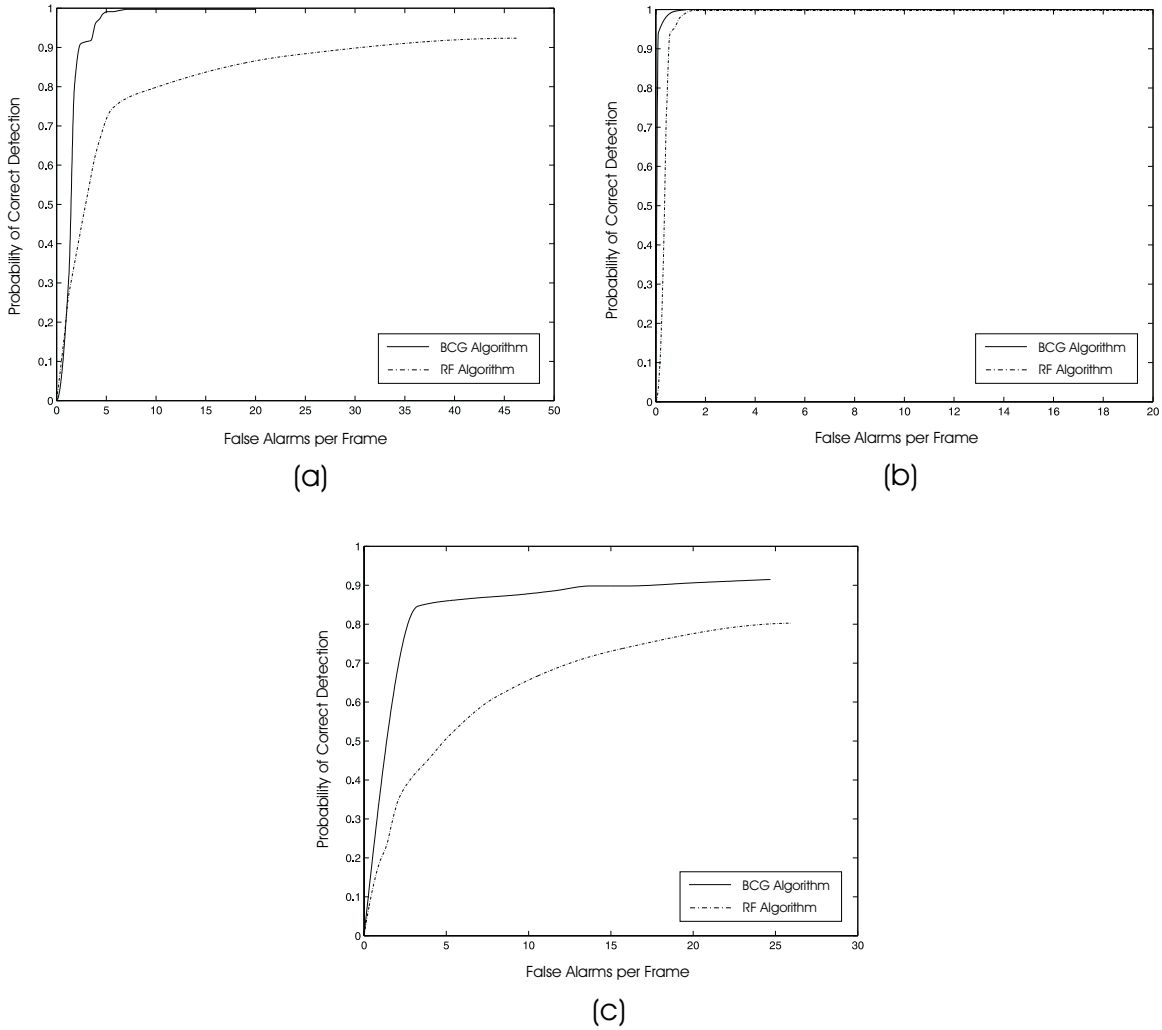


Fig. 11. Typical ROC curves obtained for the BCG and the RF algorithms and for three representative FLIR video sequences with: (a) low-contrast frames and target illumination that is similar to the background, (b) high-contrast frames and low-cluttered background, and (c) highly-cluttered frames with nonuniform contrast.

algorithm for the same number of false alarms per frame. Moreover, the ROC curve of the BCG algorithm reaches the saturation level much faster than the RF Algorithm with few false alarms per frame. Finally, Fig. 11(c) depicts the ATDT performance obtained for a case of heavily cluttered frames and nonuniform contrast. In this particular case, the video sequence is quite complex. The background contains dust, smoke, and other undesirable objects, and its contrast is highly nonuniform, which makes detection of actual targets difficult. The depicted results clearly indicate that the ATDT performance of the BCG algorithm is superior to the performance of the RF algorithm, with the BCG algorithm sharply reaching about 90% correct detection with very few false alarms per frame.

The average performance of the two ATDT algorithms is summarized in Table 1. The results have been derived by applying both algorithms on 38 FLIR sequences. The average number of frames per sequence is 355, with the longest sequence containing 779 frames and

Table 1. Average ATDT performance.

False Alarms per Frame	Probability of Correct Detection (%)		
	BCG Algorithm	RF Algorithm	% Difference
1	67.94	45.87	48
2	82.09	53.54	53
3	86.33	58.34	48
5	91.12	63.78	43
10	94.45	71.96	31
15	95.64	76.55	25
20	96.14	79.86	20

the shortest sequence containing 130 frames. Note that the average ATDT performance of the BCG algorithm ranges from 67.94% to 96.14% correct detection, with 1 to 20 false alarms per image, respectively. On the other hand, the average ATDT performance of the RF algorithm ranges from 45.87% to 79.86% correct detection. Moreover, for a fixed number of false alarms per frame, the average ATDT performance of the BCG algorithm is from 20% to 53% better than the average ATDT performance of the RF algorithm. The BCG algorithm clearly displays substantially better ATDT performance than the RF algorithm at low false alarm rates.

5 Conclusion

We have presented a method for ATDT in FLIR video, based on morphological connected operators. The proposed method avoids complications due to target and clutter variability, which are typical to FLIR ATDT, by employing general size, connectivity and motion criteria. An important aspect of the proposed scheme is that it makes no modelling assumptions and requires knowledge of only a few parameters. Another feature is that it detects both “hot” and “cold” targets with the same efficiency.

All parameters, except a threshold parameter, can be determined from available data. Once determined, these parameters remain fixed for the entire image sequence. The threshold parameter is estimated adaptively for each frame. Due to adaptive double thresholding, the performance of our algorithm is very robust to variations in illumination. Moreover, the proposed scheme is very well suited to pipelined processing or time progressive coding/transmission. Hence, it can be potentially implemented in real time.

We have demonstrated the effectiveness of the proposed algorithm by applying it on real FLIR data, which are characterized by substantial target and clutter variability. Our method shows a great deal of robustness and produces excellent detection results.

Acknowledgments

This work was supported by the Office of Naval Research, Mathematical, Computer, and Information Sciences Division, under ONR Grants N00014-90-1345 and N00014-01-1-0027. The first author was also supported by the CNPq Scholarship 200725196-3 of the Brazilian government. The authors would like to thank Dr. Richard Sims of the U.S. Army Missile Command (MICOM) for providing the data presented in this paper, and Dr. Michael I. Miller of the Center for Imaging Science for fruitful discussions.

References

- Braga-Neto, U. M., Goutsias, J., 2003. A theoretical tour of connectivity in image processing and analysis. *Journal of Mathematical Imaging and Vision* 19, 5–31.
- Braga-Neto, U. M., 2001. Connectivity in image processing and analysis: Theory, multiscale extensions, and applications. Ph.D. thesis, Center for Imaging Science and Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, Maryland.
- Goutsias, J., Batman, S., 2000. Morphological methods for biomedical image analysis. In: Sonka, M., Fitzpatrick, J. M. (Eds.), *Handbook of Medical Imaging. Volume 2. Medical Image Processing and Analysis*. SPIE Press, Bellingham, Washington, pp. 175–272.
- Heijmans, H. J. A. M., 1994. *Morphological Image Operators*. Academic Press, Boston, Massachusetts.
- Lanterman, A. D., Miller, M. I., Snyder, D. L., 1997. General Metropolis-Hastings jump diffusions for automatic target recognition in infrared scenes. *Optical Engineering* 36 (4), 1123–1137.
- Rivest, J.-F., Fortin, R., 1996. Detection of dim targets in digital infrared imagery by morphological image processing. *Optical Engineering* 35 (7), 1886–1893.
- Salembier, P., Oliveras, A., Garrido, L., 1998. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing* 7 (4), 555–570.
- Salembier, P., Serra, J., 1995. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing* 4 (8), 1153–1160.
- Serra, J., Salembier, P., 1993. Connected operators and pyramids. In: *Proceedings of the SPIE Conference on Image Algebra and Morphological Image Processing IV*. vol. 2030, San Diego, California, pp. 65–76.
- Serra, J. (Ed.), 1988. *Image Analysis and Mathematical Morphology. Volume 2: Theoretical Advances*. Academic Press, London, England.
- Serra, J., 1998. Connectivity on complete lattices. *Journal of Mathematical Imaging and Vision* 9, 231–251.
- Soille, P., 1999. *Morphological Image Analysis: Principles and Applications*. Springer, Berlin, Germany.
- Vincent, L., 1993. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing* 2 (4), 176–201.