# INCLUSION FILTERS: A CLASS OF SELF-DUAL CONNECTED OPERATORS

Nilanjan Ray, *Member*, *IEEE*, and Scott T. Acton, *Senior Member*, *IEEE*

*Abstract*– In this paper we define a connected operator that either fills or retains the holes of the connected sets depending on application-specific criteria that are increasing in the set theoretic sense. We refer to this class of connected operators as inclusion filter, which is shown to be increasing, idempotent and self-dual (gray level inversion invariance). We demonstrate self-duality for 8-adjacency on a discrete Cartesian grid. Inclusion filters are defined first for binary-valued images, and then the definition is extended to grayscale imagery. It is also shown that inclusion filters are levelings, a larger class of connected operators. Several important applications of inclusion filters are demonstrated – automatic segmentation of the lung cavities from magnetic resonance imagery, user interactive shape delineation in content based image retrieval, registration of intravital microscopic video sequences, and detection and tracking of cells from these sequences. The numerical performance measures on 100 cell tracking experiments show that the use of inclusion filter improves the total number of frames successfully tracked by five times and provides a threefold reduction in the overall position error.

*Index terms*– connected operator, adjacency tree, self-duality, level sets.

Accepted in: IEEE Transactions on Image Processing (EDICS 2-NFLT)
This work has been supported in part by the Whitaker Foundation.
Corresponding Author:

Scott T. Acton
Dept. of Electrical and Computer Engineering
351 McCormick Road
University of Virginia
Charlottesville, Virginia 22904
Phone (434) 982-2003, Fax (434) 924-8818, acton@virginia.edu

## I. INTRODUCTION

An important class of nonlinear filters is connected operators [7],[8],[13],[21],[24]. An operator is called connected if it deletes or retains (but does not distort) the connected components of a binary image and/or the connected components of the complement of a binary image [20],[21]. One of the ways to define such filtering is to decompose a grayscale image into level sets (binary images obtained by thresholding the grayscale image), filter the level sets, and then re-combine the filtered level sets [20],[21]. This framework of filtering is characterized by contrast invariance and a number of other desirable properties, such as edge localization, scale-space causality, and Euclidean invariance [1]. Examples include flat zone filters, area open, area close, and grain operators [8].

Contrast invariant connected operators have played important roles in many image processing applications. To segment an image comprised of objects of different scales or sizes, Acton and Mukherjee generate an image scale-space through images obtained at different scales using the area open-close operator, and then apply clustering within the image scale-space [1]. Document page segmentation is an important application based on the same idea of clustering within such scale-spaces [16]. Masnou and Morel propose joining missing level lines (boundaries of connected components in an image level set) to remove disocclusions from images [12]. Breen and Jones propose attribute openings with general increasing criteria in the set theoretic sense for filtering connected components [3]. Wilkinson and Westenberg apply the shape granulometry, a shape scale-space, to analyze and extract thin filaments [25]. Monasse considers matching connected components for image registration [14]. Caselles *et al*. put forward the idea of local histogram modification within the framework of image level sets [5], and Salembier *et al*. apply the anti-extensive framework of connected operators to image sequence processing [20]. A further

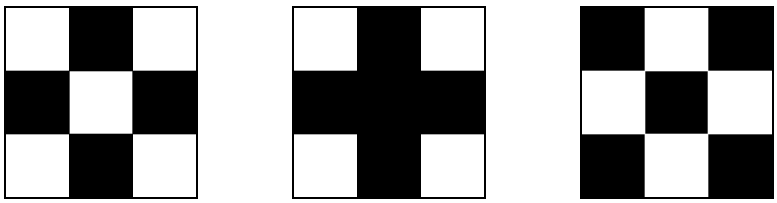
development along this trend has been contributed by Masnou with the introduction of the grain filter within the framework of self-dual connected operators [6],[11]. This filter does not behave differently under dark/bright inversion as does the area morphological operator of Salembier and Serra [21]. More general classes of connected operators, *viz*., levelings and flattenings have also been introduced [13].

The proposed inclusion filter is a class of connected operators defined on the discrete Cartesian domains to achieve perfect self-duality. Next section illustrates necessary background for inclusion filters with the aid of adjacency tree for binary images. Section III formally defines inclusion filters. Section IV shows that inclusion fitters are idempotent, increasing, and self-dual. Section V shows similarities and differences of the proposed filter with other classes of connected operators. Section VI provides example applications for inclusion filters, and Section VII indicates future research plans and conclusions to this work.

## II. BACKGROUND

The topographic map or the level line (boundary of a connected component within a level set) based representation of a grayscale image on a continuous domain fosters a class of contrast invariant connected operators that fill the holes of connected sets [6]. The key concept is that a closed level line forms a hole and that level lines are nested and non-intersecting. Due to the existence of Jordan curve theorem for discrete domain imagery, the concept of hole filling can be directly adapted to the discrete Cartesian domain if one considers 4-8 or 8-4 adjacency [15] (by *n*-*m* adjacency for a binary image we denote *n*-adjacency for the foreground and *m*-adjacency for the background on the discrete Cartesian domain). But, such direct adaptation of a level line based approach in the discrete Cartesian domain results in non-self-dual filters.

An operator $T$ is self-dual if for any binary image, $L$: $T[L] = \overline{T[\overline{L}]}$. As an example, Figure 1(a) shows a checkerboard. Let us assume 8-adjacency for foreground (black) and 4-adjacency for background (white). Note that under such an assumption all five white squares are not connected to each other, and they represent five connected components of the background. On the other hand, since black squares are 8-adjacent, four black squares from a single connected component of the foreground. In fact, this foreground connected component defines a level line enclosing a background connected component (the central white square). Thus the only hole in Figure 1(a) is the central white square. Assuming each square is of unit area, if we want to fill all holes with area less than 2, then we obtain Figure 1(b). Figure 1(c) shows the complement of Figure 1(a). Note that there is no hole in Figure 1(c) with respect to 8-4 adjacency. Thus filtering Figure 1(c), with the same hole-filling criterion, results in Figure 1(c) itself. Now the complement of Figure 1(c) is Figure 1(a), which is not the same as Figure 1(b). So the grain filters [15] implemented on the discrete Cartesian domain are not self-dual.



(a) (b) (c)

Figure 1 (a) Checkerboard $L$ and $\overline{T[\overline{L}]}$. . (b) $T[L]$. (c) $\overline{L}$ and $T[\overline{L}]$. $T[L] \neq \overline{T[\overline{L}]}$.

In this paper, we consider traditional graph theoretic connectivity (with 4- and 8-adjacency on $\mathbf{Z}^2$). Other connectivity classes are discussed in [2] and [22]. Instead of relying on the level line based description for holes on discrete Cartesian domain, we show that the adjacency tree [8],[19] approach results in a perfectly self-dual filter, provided we consider only 8-adjacency for both

foreground and background of a binary image. Let us first illustrate the adjacency tree of a binary image. Figure 2(a) shows a binary image, where we label the connected components of foreground and those of the background, respectively in black and white. The adjacency forest (collection of adjacency trees) for Figure 2(a) is shown in Figure 2(d). The white and the black nodes of the adjacency tree respectively denote the connected components of background and those of foreground. The root nodes in the forest designate the connected sets touching the image boundary. Rosenfeld proves the existence of adjacency trees in binary images when 4-8 or 8-4 adjacency is considered [19]. We define inclusion filters based on the 8-8 adjacency; we give a proof for the existence of 8-8 adjacency trees in the Appendix.

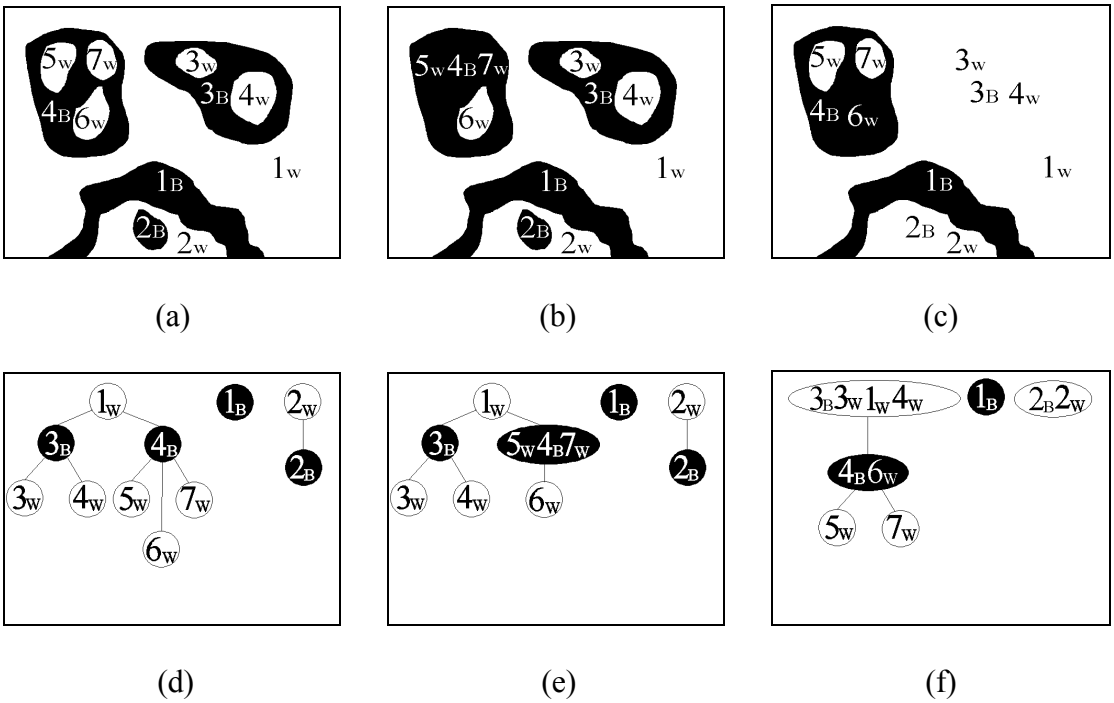



Figure 2 (a) A binary image with labeled foreground (black) and background (white) connected components. (b) Filling the two holes $_{5_w}$ and $_{7_w}$ in (a). (c) Result of an inclusion filter on (a). (d),(e) and (f) Adjacency forest for respectively (a),(b) and (c).

As mentioned, on discrete domains, we desire to define holes based on the adjacency trees of binary images, but not on level lines. To illustrate the concept of holes by means of an adjacency

tree, let us consider the connected set $1_W$ in Figure 2(a). The two holes of $1_W$ are $3_B \cup 3_W \cup 4_W$ and $4_B \cup 5_W \cup 6_W \cup 7_W$. Similarly, the only hole of the connected set, $2_W$, is $2_B$. With the help of the adjacency tree we can also demonstrate the filling of a hole. Consider filling the two holes, $5_W$ and $7_W$ of $4_B$. The binary image and the corresponding adjacency forest after filling these holes are shown respectively in Figures 2(b) and 2(e). This leads to the proposed filter we refer to as inclusion filter, the informal description for which is as follows:

*Retain or fill the holes of connected sets based on some criterion, which is an increasing function, mapping sets of points to the elements of the set {0,1}, where 0 means 'fill' and 1 means 'retain.'*

The inclusion filter paradigm may be viewed as a class of perfectly self-dual connected filters. An example of increasing criterion defining an inclusion filter is as follows: retain a hole if it intersects either the set $5_W$ or the set $7_W$; otherwise, fill the hole. Filtering the image in Figure 2(a) with this criterion, we obtain the result shown in Figure 2(c), the corresponding adjacency forest is shown in Figure 2(f).

## III. INCLUSION FILTERS

The existence of adjacency tree for any binary image establishes a very important fact – the partial order relation of "surroundedness" – if *X*, a connected set of *S*, is adjacent to *Y*, a connected set of the complement of *S*, then either *X* surrounds *Y* (*Y* is a child node to *X* in the adjacency tree) or *Y* surrounds *X* (*X* is a child node to *Y* in the adjacency tree), where *S* is any set of points in the 2-D lattice [10],[19]. Therefore, if a foreground (background) connected set is finite, then a unique background (foreground) connected set surrounds it [19]. Note that in terms of adjacency tree, surroundedness is equivalent to the parent-child relationship among the nodes.

Let $S$ be a node in the adjacency forest. We will denote by $F(S)$ the set of points belonging to the sub-tree rooted at $S$. A point is called an image boundary point when at least one of its 8-neighbors does not belong to the image domain. We can now formally characterize holes and filling of holes.

*Holes of a Connected Set*

If $C_1, C_2, \ldots, C_n$ are the children nodes of $S$, then we say $S$ has $n$ holes, and they are $F(C_1), F(C_2), \ldots, F(C_n)$. Any unbroken path (a set of ordered adjacent points) from a point belonging to a hole, $F(C_i)$ of $S$, to any point belonging to the image boundary, must intersect $S$. Conversely, if all unbroken paths from a point to the image boundary always intersect $S$, then the point must belong either to $S$ or to some hole, $F(C_i)$ of $S$. Any root node in the adjacency forest is called an *infinite connected set*. Thus, by definition no hole can be an infinite connected set.

*Filling of a Hole*

Let $S$ be of color black (white) in the adjacency forest. Filling a hole, $F(C_i)$ of $S$, means to change the color of the points belonging to $F(C_i)$ to black (white), *i.e.*, the same color as the points of $S$. In other words, filling the hole $F(C_i)$ means to merge the sub-tree rooted at $C_i$ with $S$. After filling the hole, $F(C_i)$, the node $S$ of the old adjacency forest, will appear as $S \cup F(C_i)$ in the new adjacency forest. Similarly, filling all the holes of $S$ will result in an adjacency tree where, the set $F(S) = S \cup F(C_1) \cup \cdots \cup F(C_n)$ will appear as a node with the same color as $S$.

Based on the structure of adjacency forest for a binary image, we can characterize an inclusion sequence as follows.

*Inclusion Sequence*

Let $L$ be a binary image and **x** be a point in the image domain. The following holds:

(i) if $L(\mathbf{x}) = 1$, then there exists a unique alternating sequence of foreground and background connected sets – $C_1$, $H_2$, $C_3$,…, such that $\mathbf{x} \in C_1$ and in the adjacency tree $H_2$ is the parent of $C_1$, $C_3$ is the parent of $H_2$, and so on.

(ii) if $L(\mathbf{x}) = 0$, then there exists a unique alternating sequence of background and foreground connected sets – $H_1$, $C_2$, $H_3$, …, such that $\mathbf{x} \in H_1$ and in the adjacency tree $C_2$ is the parent of $H_1$, $H_3$ is the parent of $C_2$, and so on. Such a sequence will henceforth be called an inclusion sequence (IS). Any IS always terminates with an infinite connected set.

Next, we define an increasing criterion.

*Increasing Criterion*

Let $\Omega$ be the image domain, then a function, $T$, mapping the power set of $\Omega$ to the set $\{0,1\}$, is increasing if $0 \le T(S_1) \le T(S_2) \le 1$, for any $S_1 \subset \Omega, S_2 \subset \Omega$ with $S_1 \subset S_2$. Additionally we stipulate that $T(S) = 1$ for any infinite connected set $S$.

Finally we define inclusion filters with the use of IS and increasing criteria as follows.

*Inclusion Filtering*

Let $C_1, H_2, C_3 \ldots$ or $H_1, C_2, H_3 \ldots$ be an IS for **x** in the binary image $L$, where $C$'s and $H$'s respectively denote foreground and background connected sets. Now, $F(C_n)$ is a hole of $H_{n+1}$, $F(H_{n+1})$ is a hole of $C_{n+2}$, and so on. So, we have: $\ldots \subset F(C_{n-1}) \subset F(H_n) \subset F(C_{n+1}) \subset \ldots$ from

the adjacency forest. For an increasing criterion, $T$, we then have: $0 \le \ldots \le T(F(C_{n-1})) \le T(F(H_n)) \le T(F(C_{n+1})) \le \ldots \le 1$. Since the last term in this sequence is unity-valued, only the four mutually exclusive and exhaustive possibilities exist:

a) $T(F(C_1)) = T(F(H_2)) = T(F(C_3)) = T(F(H_4)) = \ldots = 1$,

b) $\exists n \ge 1$ such that $\ldots = T(F(C_{n-1})) = T(F(H_n)) = 0$ and $T(F(C_{n+1})) = T(F(H_{n+2})) = \ldots = 1$,

c) $T(F(H_1)) = T(F(C_2)) = T(F(H_3)) = T(F(C_4)) = \ldots = 1$,

d) $\exists n \ge 1$ such that $\ldots = T(F(H_{n-1})) = T(F(C_n)) = 0$ and $T(F(H_{n+1})) = T(F(C_{n+2})) = \ldots = 1$.

We define, $L_f(\mathbf{x})$, the inclusion filtered $L(\mathbf{x})$, as follows:

$$L_f(\mathbf{x}) = \begin{cases} 1, \text{ if (a) or (b) is true,} \\ 0, \text{ if (c) or (d) is true.} \end{cases}$$

In short we say $L_f(\mathbf{x}) = \psi_T(L(\mathbf{x}))$. It is straightforward to note that, in case (a) and (c), $L_f(\mathbf{x}) = L(\mathbf{x})$, and in case (b), the hole $F(H_n)$ of $C_{n+1}$ is filled. Similarly in case (d), the hole $F(C_n)$ of $H_{n+1}$ is filled up. Thus, inclusion filter is a formal description of filling the holes in a binary image based on increasing criteria defined on the holes. An inclusion filter assigns the color black or white possibly to more than one node in a single branch of an adjacency tree. If we wanted to define a filter that deletes one leaf node at a time depending on an increasing criterion, then we would not obtain an idempotent operator, because several leaf node deletions may be performed for a particular increasing criterion.

## IV. PROPERTIES OF INCLUSION FILTERS

We discuss three important properties of inclusion filters and prove their existence from digital topological standpoint. (Digital topology is the study of connectivity on discrete domains [10],[19].) We show that inclusion filters are idempotent, increasing, and self-dual. A filter is known as idempotent when the output of the filter is not changed by subsequent application of the same filter [23]. This property makes a filter non-iterative in nature. Increasing nature of a filter, on the other hand, is very much desirable, as it widens the applications of the filter from the binary images to grayscale images [23]. Self-duality of a filter makes it unbiased towards bright and dark objects in an image [23]. Self-duality is crucial in applications such leukocyte tracking/detection from intravital video imagery illustrated in Section IV.

### *A. Idempotency*

To prove the idempotency of the inclusion filter we first state and prove the following lemma.

**Lemma I.** Let $S_1$ and $S_2$ be two connected sets such that $S_1 \supset S_2$, then $F(S_1) \supset F(S_2)$.

**Proof.** Let $\mathbf{p} \in F(S_2)$, *i.e.*, **p** belongs either to $S_2$ or to a hole of $S_2$. Then any unbroken path, $\pi$, from **p** to the image border, intersects $S_2$, *i.e.*, $\pi \cap S_2 \neq \phi$. Then, $\pi \cap S_1 \neq \phi$, as $S_1 \supset S_2$. But $\pi$ is an arbitrarily chosen path from **p** to the image border. Thus, any path from **p** to the image border intersects $S_1$, and **p** must belong either to $S_1$ or to a hole of $S_1$, *i.e.*, $\mathbf{p} \in F(S_1)$. Therefore $F(S_1) \supset F(S_2)$. ■

**Proposition I.** Inclusion filters are idempotent.

**Proof.** Let $L_f(\mathbf{x}) = \psi_T(L(\mathbf{x}))$. We first consider case (a). After filtering, the IS of **x** in $L_f$ becomes $C_1^{'}, H_2^{'}, C_3^{'}, \ldots,$ with $C_1^{'}, \supset C_1, H_2^{'} \supset H_2, C_3^{'} \supset C_3, \ldots.$ Therefore, by Lemma I: $F(C_1^{'}) \supset F(C_1), F(H_2^{'}) \supset F(H_2), F(C_3^{'}) \supset F(C_3), \ldots.$ Thus, $T(F(C_1^{'})) = T(F(H_2^{'})) = T(F(C_3^{'})) = \ldots = 1,$ so $L_f(\mathbf{x}) = \psi_T(L_f(\mathbf{x}))$, implying that the filter is idempotent in this case. In case (b), the hole $F(H_n)$ of $C_{n+1}$ is filled. So after the inclusion filtering the IS for **x** becomes $C_1^{'}, H_2^{'}, C_3^{'}, \ldots,$ with $C_1^{'}, \supset C_{n+1}, H_2^{'} \supset H_{n+2}, C_3^{'} \supset C_{n+3},$ and hence, the same result follows. Cases (c) and (d) are similar. ■

### *B. Increasingness and Extensibility to Grayscale Imagery*

We prove that binary inclusion filters are increasing – if $L_1(\mathbf{x})$ and $L_2(\mathbf{x})$ are two binary images with $L_1(\mathbf{x}) \geq L_2(\mathbf{x}), \forall \mathbf{x} \in \Omega$, then: $L_{1f}(\mathbf{x}) \geq L_{2f}(\mathbf{x}), \forall \mathbf{x} \in \Omega,$ where $L_{1f}$ and $L_{2f}$ are inclusion filter outputs of $L_1$ and $L_2$, respectively. The increasing nature of any binary filter allows applicability to grayscale imagery: the input grayscale image is first threshold decomposed into binary images (level sets) with increasing thresholds, then each of these binary images is filtered with inclusion filter, and finally the output binary images are stacked to obtain the output grayscale image [20],[21],[23]. The following algorithm extends a binary inclusion filter to a grayscale image.

**Algorithm I**

(a) Decompose the grayscale image, *I*, into level sets: $L^{\lambda}(\mathbf{x}) \leftarrow \begin{cases} 1 \text{ if } I(\mathbf{x}) \geq \lambda \\ 0 \text{ otherwise.} \end{cases}$

(b) Inclusion filter each level set: $L_f^{\lambda}(\mathbf{x}) = \psi_T(L^{\lambda}(\mathbf{x}))$.

(c) Reconstruct the filtered image, $I_f$, by stacking the filtered level sets $L_f^\lambda$: $I_f(\mathbf{x}) \leftarrow \max\{\lambda : L_f^\lambda(\mathbf{x}) = 1\}$.

Note that the threshold decomposition principle in Algorithm I inherently restricts that increasing criteria be defined on the holes of an image level set, but not on grayscale properties, such as intensity. Also Algorithm I is meaningful when a binary inclusion filter is increasing. To show the increasing nature of inclusion filters, we need Lemma II.

**Lemma II.** Let $S_1$ and $S_2$ be two connected sets with holes $H_1^1, H_2^1, \ldots$ and $H_1^2, H_2^2, \ldots$, respectively. Let $S_1 \supset S_2$ and let **x** be a point, such that $\mathbf{x} \in F(S_2)$ and $\mathbf{x} \in H_i^1$, for some *i*. Then there exists $H_j^2$, for some *j*, such that $H_j^2 \supset H_i^1$.

**Proof.** Let any arbitrary $\mathbf{p} \in H_i^1$ and $\mathbf{p} \neq \mathbf{x}$. Given that $\mathbf{x} \in H_i^1$, there exists an unbroken path $\pi$ from **x** to **p**, such that $\pi \subset H_i^1$. Again $H_i^1 \cap S_1 = \phi$, so we obtain the result: $\pi \cap S_1 = \phi$, which we refer to as R1. Now $\mathbf{x} \in H_i^1 \Rightarrow \mathbf{x} \notin S_1$. Therefore $\mathbf{x} \notin S_2$ as $S_1 \supset S_2$. But it is given that $\mathbf{x} \in F(S_2)$, so there must exist a hole, $H_j^2$ of $S_2$, such that $\mathbf{x} \in H_j^2$. If we assume: $\mathbf{p} \in \overline{F(S_2)}$, then the path, $\pi$ from **x** to **p**, must intersect $S_2$, *i.e.*, $\pi \cap S_2 \neq \phi$. Since $S_1 \supset S_2$, we have: $\pi \cap S_1 \neq \phi$, which is contradictory to R1. Thus, our assumption $\mathbf{p} \in \overline{F(S_2)}$ is false, so $\mathbf{p} \in F(S_2)$. Now $\mathbf{p} \in H_i^1 \Rightarrow \mathbf{p} \notin S_1$. Then $S_1 \supset S_2 \Rightarrow \mathbf{p} \notin S_2$. Since $\mathbf{p} \in F(S_2)$ and $\mathbf{p} \notin S_2$, there must exist a hole $H_l^2$ of $S_2$, such that $\mathbf{p} \in H_l^2$. Let us now assume that $l \neq j$. Since $S_2$ surrounds both $H_j^2$ and $H_l^2$ and since $\mathbf{x} \in H_j^2$ and $\mathbf{p} \in H_l^2$, it follows that $\pi \cap S_2 \neq \phi$, which is contradictory to R1, so $l = j$.

Therefore for any arbitrary $\mathbf{p} \neq \mathbf{x}$, $\mathbf{p} \in H_i^1 \Rightarrow \mathbf{p} \in H_j^2$. Now we know that $\mathbf{x} \in H_i^1$ and $\mathbf{x} \in H_j^2$. Therefore $\mathbf{p} \in H_i^1 \Rightarrow \mathbf{p} \in H_j^2, \forall \mathbf{p}$. So $H_j^2 \supset H_i^1$. ■

Now we prove that binary inclusion filters are increasing.

**Proposition II.** $L_1(\mathbf{x}) \geq L_2(\mathbf{x}),\ \forall \mathbf{x} \in \Omega \Rightarrow L_{1f}(\mathbf{x}) \geq L_{2f}(\mathbf{x}),\ \forall \mathbf{x} \in \Omega$.

**Proof.** It suffices to prove that $L_{2f}(\mathbf{x}) = 1 \Rightarrow L_{1f}(\mathbf{x}) = 1$, for any $\mathbf{x} \in \Omega$. Let the IS for a point **x** in $L_2$, be $C_1^2, H_2^2, C_3^2 \ldots$, or $H_1^2, C_2^2, H_3^2 \ldots$, where *C*'s and *H*'s denote respectively foreground and background connected sets. Since $L_1 \geq L_2$, for any foreground connected set, $C_n^2$ in $L_2$, there exists a foreground connected set, $C^1$ in $L_1$, such that $C^1 \supset C_n^2$. So by Lemma I we have $F(C^1) \supset F(C_n^2)$. Thus $\mathbf{x} \in F(C_n^2) \Rightarrow \mathbf{x} \in F(C^1)$. Therefore if $C_1^1, H_2^1, C_3^1 \ldots$ or $H_1^1, C_2^1, H_3^1 \ldots$ is the IS of **x** in $L_1$, then there exists $C_m^1$, such that $C_m^1 = C^1$. Now, $L_{2f}(\mathbf{x}) = 1$; so there are two cases to consider:

Case (1): $T(F(C_1^2)) = 1$. By the preceding argument there exists $C_m^1$ in the IS for **x**, and $C_m^1 \supset C_1^2$. Since $L_1 \geq L_2$, $\mathbf{x} \in C_1^2 \Rightarrow \mathbf{x} \in C_m^1$, *i.e.*, $m = 1$. Thus, by Lemma I we have $F(C_1^1) \supset F(C_1^2)$, and, by the increasing nature of *T*, we have: $1 = T(F(C_1^2)) \leq T(F(C_1^1))$, implying $T(F(C_1^1)) = 1$, and hence $L_{1f}(\mathbf{x}) = 1$ .

Case (2): $T(F(H_n^2)) = 0$ and $T(F(C_{n+1}^2)) = 1$ for some $n \geq 1$. We have already shown: $\exists C_{m+1}^1 \supset C_{n+1}^2$. Two cases are possible here. In the first case, $m = 0$, *i.e.*, the IS of **x** in $L_1$ starts with $C_1^1$. Then, $1 = T(F(C_{n+1}^2)) \leq T(F(C_1^1))$, and thus, $L_{1f}(\mathbf{x}) = 1$ . In the second case, $m > 0$,

*i.e.*, there exists $H_m^1$ in the IS of **x** in $L_1$. Then $F(H_m^1)$ is a hole in $C_{m+1}^1$ containing **x**. Therefore by Lemma II, there exists a hole, $H^2$ in $C_{n+1}^2$, such that $H^2 \supset F(H_m^1)$ and $\mathbf{x} \in H^2$. Now, both $F(H_n^2)$ and $H^2$ are holes of $C_{n+1}^2$ such that $\mathbf{x} \in F(H_n^2)$ and $\mathbf{x} \in H^2$, which is not possible unless $H^2 = F(H_n^2)$. Therefore, the increasing nature of $T$ again implies that $T(F(H_m^1)) \le T(F(H_n^2)) = 0$ and $1 = T(F(C_{n+1}^2)) \le T(F(C_{m+1}^1))$. This leads to $L_{1f}(\mathbf{x}) = 1$ . ■

It is now straightforward to prove that grayscale inclusion filters defined via Algorithm I are also increasing. If $I_1$ and $I_2$ are two grayscale images, and $I_1 \ge I_2$, then $L_1^\lambda \ge L_2^\lambda \ \forall \lambda$, where $L$'s are the level set of the respective images at the gray level $\lambda$. Now $L_1^\lambda \ge L_2^\lambda \ \forall \lambda \Rightarrow L_{1f}^\lambda \ge L_{2f}^\lambda \ \forall \lambda$, where $L$'s with subscript $f$ denote the outputs of inclusion filter. Therefore, $I_1 \ge I_2 \Rightarrow I_{1f} = \max\{\lambda : L_{1f} = 1\} \ge \max\{\lambda : L_{2f} = 1\} = I_{2f}$.

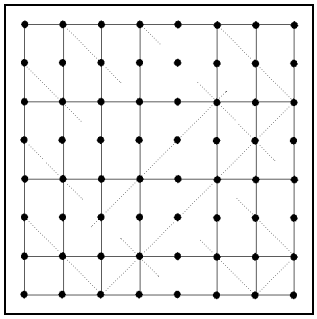


Figure 3. Khalimsky's connectivity grid [10].

*C. Self-duality*

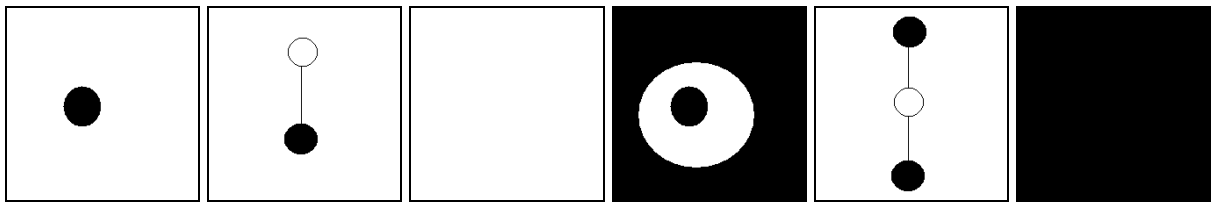
One way to achieve perfect self-duality for a connected operator is to use hexagonal grid instead of the Cartesian grid since the hexagonal grid has only one kind of adjacency viz., 6-adjacency [10]. Since the Cartesian grid is more common in use than the hexagonal grid, we may use Khalimsky's connectivity [9] on the Cartesian grid to achieve self-duality. Figure 3 shows

Klaimsky's connectivity grid. One problem posed by Khalimsky's grid is that the connectivity is not translation invariant [10]. The translation variance is a major limitation of the inclusion filter using Khalimsky's connectivity for applications such as image sequence processing.

To make the self-duality property applicable to Cartesian grid, and further to use it for image sequence processing, *e.g.*, in tracking cells, we propose to use 8-adjacency for both the foreground and the background connected sets. Achieving self-duality with such simple means is one of our contributions to inclusion filters. We claim with the help of adjacency tree that such adjacency consideration yields perfectly self-dual inclusion filters with all other properties preserved.

**Proposition III.** Inclusion filters are self-dual.

**Proof.** Under the intensity reversal of a binary image, every node in the 8-8 adjacency forest changes its color from black to white or *vice versa*, but the entire forest remains unchanged in terms of its structure. This implies that if the IS of **x** in a binary image $L$ is $C_1$, $H_2$,…,($H_1$, $C_2$,…) then the IS of **x** for the complement $\overline{L}$ of $L$ is $H_1$, $C_2$,…,($C_1$, $H_2$,…,), where $C$'s and $H$'s denote the foreground and the background connected sets. Therefore, from the definition of the inclusion filter: $\psi_T(L(\mathbf{x})) = \overline{\psi_T(\overline{L}(\mathbf{x}))}$. Note that if $L$ is a level set of a gray level image $I$, then $\overline{L}$ is a level set of the intensity-reversed image of $I$, so the result also extends to the grayscale images. ■

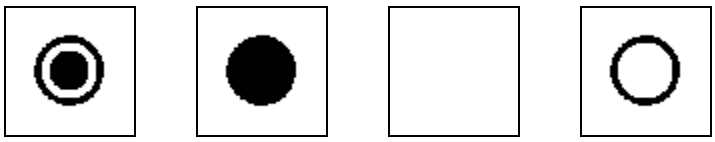


(a) (b) (c) (d) (e) (f)

Figure 4. (a) A binary image. (b) The adjacency tree for the binary image in (a). (c) Filling all holes in (a). (d) Another binary image. (e) The adjacency tree for the binary image in (d). (f) Filling all holes in (d). Leaf nodes in (b) and (d) contain same "local" information, *e.g.*, area, color, position.

## V. INCLUSION FILTERS AND OTHER CONNECTED OPERATORS

In this section we show similarities and dissimilarities of inclusion filters with other classes of connected operators such as "grain operator" defined by Heijmans [8] and levelings defined by Meyer [13]. Figures 4(a) and 4(d) show two binary images, for which the adjacency trees are shown in Figures 4(b) and 4(e). Applying an inclusion filter that fills all holes of Figures 4(a) and 4(d) results in Figures 4(c) and 4(f) respectively. For a "grain operator" the filtering result for points in a node depends only on the "local" information stored in the node (*e.g.*, the area, the color and the position of the set of points in the node), but not on any of its adjacent nodes (*e.g.*, parent or child node) [8]. Note that the leaf nodes in Figures 4(b) and 4(e) are identical, so any "grain operator" would produce the same result for the leaf nodes. But, by filling all holes, we obtain different results for the leaf nodes as shown in Figure 4(c) and 4(f). So the aforementioned inclusion filter is different from the "grain operators" of Heijmans.



(a) (b) (c) (d)

Figure 5 (a) A binary image. (b) Result of area close. (c) Result of area open. (d) Result of inclusion filter. In all three cases the same area scale of 300 is applied.

Connected operators such as area open or area close (a kind of grain operator) are notably different in principle from inclusion filters. Figure 5(a) shows a binary image. If an area threshold of 300 (for example) is applied, then area close removes all the background components with area

less than 300 (Figure 5(b)); area open then removes all foreground components with area less than 300 (Figure 5(c)), and an inclusion filter fills all holes of area less than 300 (Figure 5(d)).

However, we find that inclusion filters belong to a large class of connected operators, levelings [13].

**Proposition IV.** Inclusion filters are levelings.

**Proof.** We first prove that binary inclusion filters are levelings [13]: for any two neighbors **x** and **y**, $L_f(\mathbf{x}) > L_f(\mathbf{y}) \Rightarrow L(\mathbf{x}) \geq L_f(\mathbf{x})$ and $L_f(\mathbf{y}) \geq L(\mathbf{y})$. For the binary inclusion filter, we need to show $L_f(\mathbf{x}) = 1$ and $L_f(\mathbf{y}) = 0 \Rightarrow L(\mathbf{x}) = 1$ and $L(\mathbf{y}) = 0,$ where $L$ denotes a binary image and $L_f$ denotes the inclusion filtered output for $L$. To prove this by contradiction, we first assume: $L(\mathbf{x}) = 0,$ *i.e.*, the IS for **x** in $L$ is $H_1$, $C_2$, $H_3$, …. Now $L_f(\mathbf{x}) = 1,$ so by the definition of the inclusion filter: $\exists n \geq 1$ such that $\ldots = T(F(C_{n-1})) = T(F(H_n)) = 0$ and $T(F(C_{n+1})) = T(F(H_{n+2})) = \ldots = 1$. Note that $H_1$ is not an infinite connected set, otherwise $L_f(\mathbf{x})$ would have been same as $L(\mathbf{x})$, *i.e.*, 0. Now, **y** is a neighbor of **x**, so the IS for **y** in $L$ is either $H_1$, $C_2$, $H_3$,… or $C_2$, $H_3$,… In either case, $\ldots = T(F(C_{n-1})) = T(F(H_n)) = 0$ and $T(F(C_{n+1})) = T(F(H_{n+2})) = \ldots = 1$, and so: $L_f(\mathbf{y}) = 1,$ which is a contradiction. Therefore $L(\mathbf{x}) \neq 0.$ Similarly, assuming $L(\mathbf{y}) = 1,$ we obtain a contradiction. Thus $L(\mathbf{y}) \neq 1$. So we have shown $L(\mathbf{x}) = 1$ and $L(\mathbf{y}) = 0$ and hence binary inclusion filters are levelings.

This result can be extended to the grayscale imagery. If $L^\lambda$ and $L_f^\lambda$ are level sets of respectively the original grayscale image, $I$, and the filtered image, $I_f$, then for any two neighbors **x**, **y**: $I_f(\mathbf{x}) > I_f(\mathbf{y}) \Rightarrow L_f^\lambda(\mathbf{x}) > L_f^\lambda(\mathbf{y})$ for $I_f(\mathbf{y}) < \lambda \leq I_f(\mathbf{x})$. Now binary inclusion filter is a

leveling, so: $1 = L^{\lambda}(\mathbf{x}) \geq L_f^{\lambda}(\mathbf{x}) = 1$ for $\lambda = I_f(\mathbf{x})$. Therefore, $I(\mathbf{x}) = \max\{\lambda : L^{\lambda}(\mathbf{x}) = 1\} \geq I_f(\mathbf{x})$. Again, since the binary inclusion filter is a leveling, we have: $0 = L_f^{\lambda}(\mathbf{y}) \geq L^{\lambda}(\mathbf{y}) = 0$ for $\lambda > I_f(\mathbf{y})$, so $I_f(\mathbf{y}) \geq \max\{\lambda : L^{\lambda}(\mathbf{y}) = 1\} = I(\mathbf{y})$. ■

## VI. APPLICATIONS

### A. Marker Inclusion Filter

One basic increasing criterion for an inclusion filter can be defined based on the sets of fixed points or *markers*. Let $S_1$, $S_2$,…,$S_n$ be $n$ sets of points, *i.e.*, $S_i \subset \Omega$, where $\Omega$ is the image domain. Let $C$ be any connected set. We define the marker inclusion criterion in the following way:

$$T(F(C)) = \begin{cases} 1 \text{ if } \exists i, s.t.\ S_i \subset F(C) \\ 0 \text{ else.} \end{cases}$$

In other words, we retain a hole if the hole entirely contains at least one set of markers $S_i$; otherwise, we fill the hole. Below we give some example situations where the use of the marker inclusion filter is efficacious.

(a) (b) (c) (d)

Figure 6. (a) An image of an eagle. The user has mouse-clicked inside the head and neck of the eagle (5 black dots). (b) After inclusion filtering the eagle image with the markers (5 black dots). (c) Canny edges of (a). (d) Canny edges of (b).

Consider the user interaction phase in content-based image retrieval (CBIR): the user is interested to extract the shape of the head and neck portions of the eagle in Figure 6(a). The user chooses five marker points on the image that form the marker set, $S_1$ (5 black dots in Figure

6(a)). In Figure 6(b) we show the result of inclusion filter. Figures 6(c) and 6(d) show the Canny edges [4] of Figures 6(a) and 6(b) respectively. On the other hand, if the user had wanted to use the area open/close or grain filter with area criterion to achieve the same result, (s)he would have had to estimate the area scale on some trial and error basis, and then finally apply the filter. Thus, the inclusion filter increases the ease of user interaction in this case.

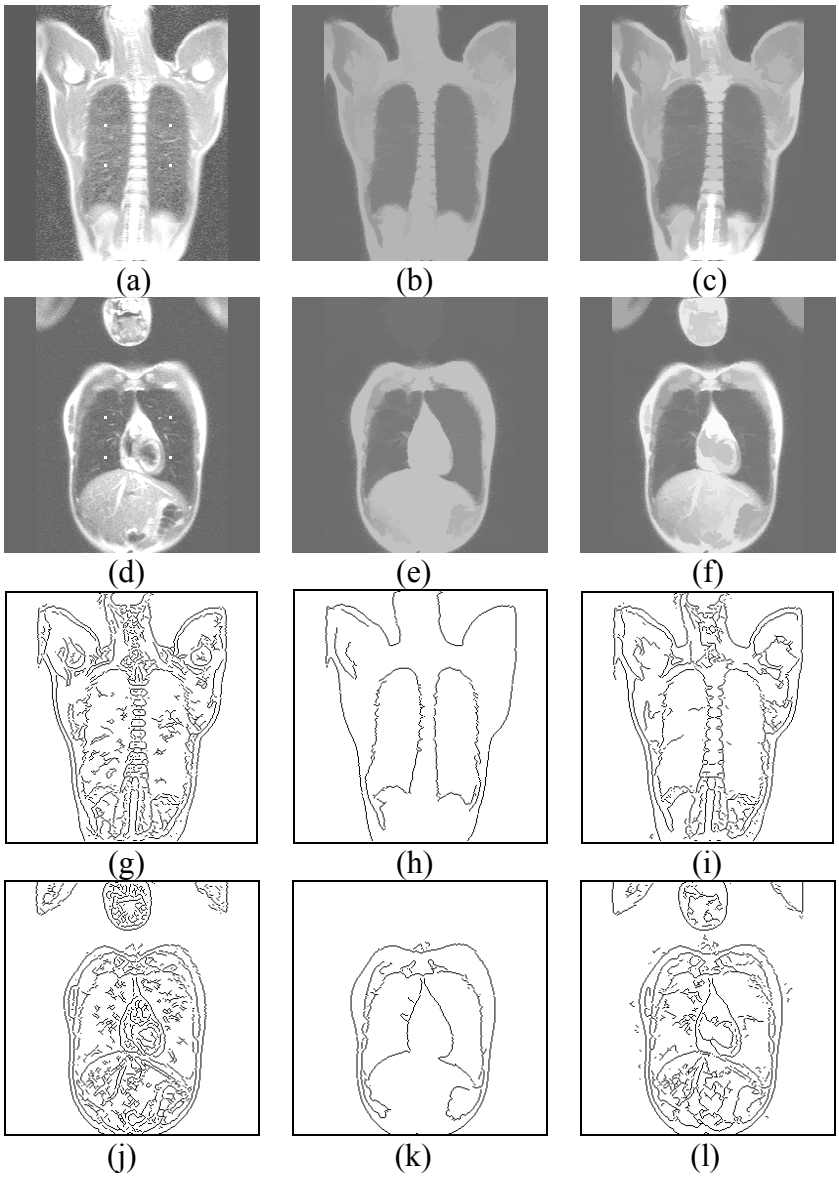



Figure 7. (a), (d) MR slices with two sets of white markers inside the lung cavities. (b), (e) Marker inclusion filter results on (a) and (d) respectively. (c), (f) Result of grain filters on (a) and (d) respectively with a same area scale of 500 pixels. (g)-(l) Canny edges of (a)-(f) in this order.

The inclusion filter can also be applied to segmenting magnetic resonance (MR) imagery [17]. Two typical MR slices of the lungs are shown in Figures 7(a) and 7(d). Their Canny edges are shown in Figures 7(g) and 7(j), where the presence of the spurious edges obscures the lung cavity edge from which the lung cavity area can be computed. Although taken from the same MR sequence, these images reveal that the area of all the anatomical details (*e.g.*, lung cavity size, surroundings of the lung cavities) vary in size. This implies that fixing an area scale for the area open/close filter or grain filter that works in all the slices in a single MR sequence would be difficult. Fixing different area scales for different MR slices is even more difficult on an operational basis, as the lung sizes will vary enormously in depth and breadth from patient to patient. However the marker inclusion filter can be easily applied to all of the MR slices once we are able to fix two sets of markers (the white points in Figure 7(a) and Figure 7(d)), one set each within each lung cavity. Figures 7(b) and 7(e) show the results of marker inclusion filter respectively on Figures 7(a) and 7(d). As a comparison Figures 7(c) and 7(f) show the results of grain filter respectively on Figures 7(a) and 7(d) with an area scale of 500 pixels. In Figures 7(g) through 7(l) we show the Canny edges of Figures 7(a) through 7(f) in that order.

B. *Shape Inscription Filter*

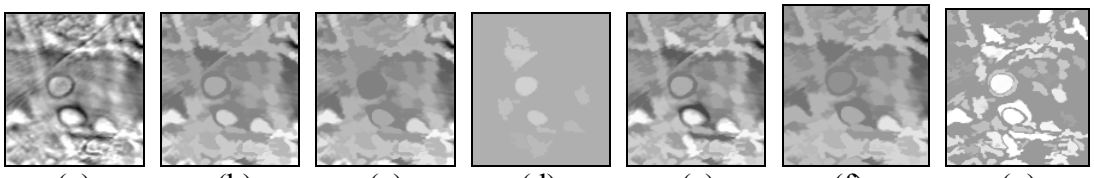
A potential drawback of the area open-close filter is that being based entirely on an area criterion, the filter ignores the shape information. So a round-shaped connected set and a long, slender connected set of the same area would be treated in the same manner in these filters. This motivates the formulation of a *shape inscription* criterion for the inclusion filter. This particular increasing criterion retains a hole if a given shape can be inscribed in it; otherwise it fills the hole. Let $S \subset Z^2$ designate a given shape. Let us further assume that we want to inscribe the shape inside the holes with all possible movements by translation and rotation of the shape. In

this connection we point out that open-by-reconstruction is a similar type of filter where a connected component is retained if the binary erosion of this connected component with a structuring element of certain size is not an empty set; otherwise the connected component is deleted [8]. It is also to be noted that here shape inscribing is performed in every hole, *i.e.*, every filled in foreground and background connected set, unlike open-by-reconstruction, where the erosion is performed only on the foreground connected sets without filling the holes in them.

We now define the shape (a set of points, $S$) inscription criterion on a connected set, $C$, as follows:

$$T(F(C)) = \begin{cases} 1 \text{ if } S \text{ can be inscribed in } F(C) \\ 0 \text{ else.} \end{cases}$$

We apply this filter to the image frames for the detection of leukocytes. Figure 8(a) shows a video frame containing leukocytes. Figures 8(b) and 8(c) show the results of shape inscription filter on Figure 8(a) with circles of radii 5 and 9 pixels respectively. Figure 8(d) shows the frame difference of Figures 8(b) and 8(c). To show the comparison with grain filtering results we have used two area scales, 79 and 255 pixels, to generate Figures 8(e) and 8(f). Since the numbers 79 and 255 correspond to the areas of circles of radii 5 and 9 we may view them as the equivalent scales in terms of the shape inscription filter used here. The difference image for Figure 8(e) and Figure 8(f) is shown in Figure 8(g). Figure 8(d) appears to have reduced clutter compared to Figure 8(g).



(a) (b) (c) (d) (e) (f) (g)

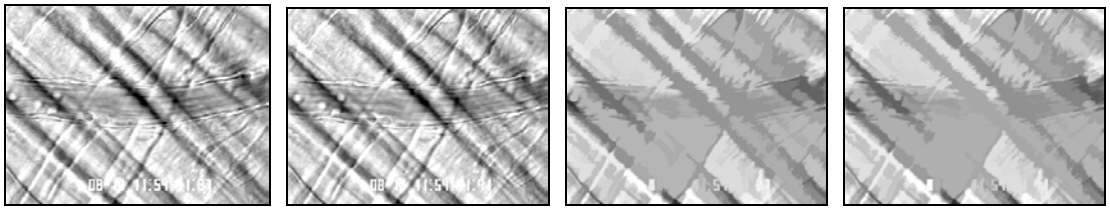
Figure 8. (a) An image containing leukocytes. (b) Shape inscription filtering with a circle of radius 5 pixels. (c) Shape inscription filtering with a circle of radius 9 pixels. (d) The difference image of (b) and (c). (e) Grain filtering at area scale 79 pixels. (f) Grain filtering at area scale 255 pixels. (g) The difference image of (e) and (f).

C. *Infinite Set Filter*

Another increasing criterion leads to filling all holes. The corresponding inclusion filter is called the *infinite set filter*. This filter applies the following criterion:

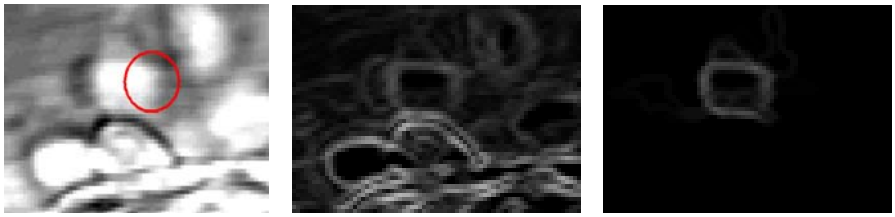
$$T_{\text{inf}}(F(C)) = \begin{cases} 1 \text{ if } C \text{ is infinite} \\ 0 \text{ otherwise.} \end{cases}$$

This filter may be applied to the problem of video frame registration. For video frames obtained *in vivo*, registration is often a strong requirement as the subject is living and moving introducing jitter in the video [18]. Two such consecutive video frames are shown in Figure 9(a) and Figure 9(b). The microvessel and leukocytes are seen through the transparent muscle tissues (in Figures 9(a) and 9(b), the dark and bright lines appearing thick as well as thin across the image). Since the muscle tissue is relatively fixed in the living subject, we may use them in registering the frames for the purpose of tracking leukocytes. Figure 9(c) and Figure 9(d) are the results of infinite set filter on Figure 9(a) and Figure 9(b), respectively. It is to be noted that the infinite set filter does not utilize any (*ad hoc* or otherwise determined) scale parameter. The filtering result depends only on the image boundary. We observe from the filtering results that the muscle striations are retained while the small artifacts and moving leukocytes are removed, thus making it possible to register them with frame correlation.



(a) (b) (c) (d)

Figure 9. (a) and (b) Two consecutive *in vivo* video frames showing blood vessel, leucytes and muscle striations. (c) and (d) Infinite set filtering on (a) and (b) respectively.



(a) (b) (c)

Figure 10. (a) Leukocytes and the set $S$ (interior of the circular contour). (b) $f = |\nabla I|$. (c) $f_{\text{filt}} = \nabla |\psi_{T_{\text{track}}}(I)| - \nabla |\psi_{T_{\text{inf}}}(I)|$.

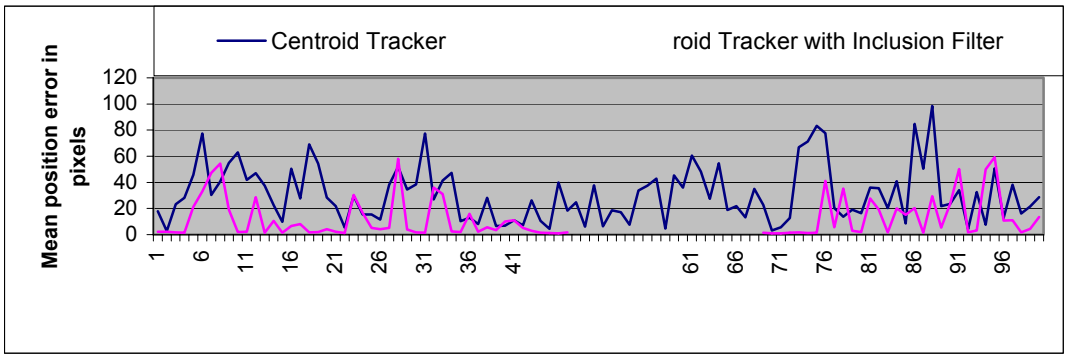
*D. Inclusion Filter for Leukocyte Tracking*

Inclusion filters can significantly enhance the performance of centroid trackers [18] for tracking cells within an intravital (in living subject) video sequence. One of the characteristics of such intravital image sequences is the intensity reversal of the cells. A bright cell in one frame may appear dark in the very next frame [18]. Thus the self-duality is ideally required for this application. Let $I$ be an image and let the corresponding edgemap be defined as $f = |\nabla I|$. With the centroid method, the cell center is computed by finding the center of mass of the edgemap within some search window. The frames obtained via video microscopy are often cluttered and noisy leading to erroneous computation of the cell center. So, before being processed by the centroid tracker, the frames are filtered with the following criterion:

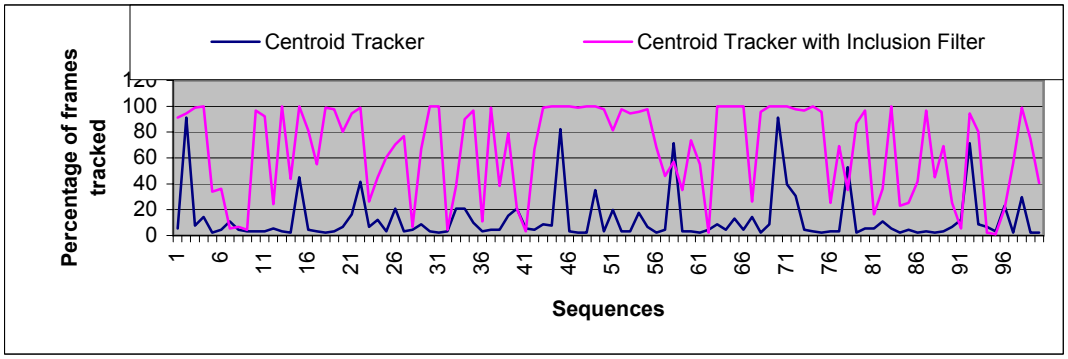
$$T_{\text{track}}(F(C)) = \begin{cases} 1 \text{ if } F(C) \cap S \neq \phi \\ 0 \text{ otherwise,} \end{cases}$$

where $S = \{(x,y) : (x-\bar{x})^2 + (y-\bar{y})^2 \leq R^2\}$ and $(\bar{x}, \bar{y})$ is the estimated cell center and $R$ is the radius of the cell, known *a priori*. To discard the effect of edges touching the frame boundary, the filtered edgemap is computed as $f_{\text{filt}} = \nabla |\psi_{T_{\text{track}}}(I)| - \nabla |\psi_{T_{\text{inf}}}(I)|$, where $\psi_{T_{\text{track}}}(I)$ and $\psi_{T_{\text{inf}}}(I)$ respectively denote the outputs of inclusion filter with the tracking criterion, $T_{\text{track}}$ and

infinite set filtering criterion, $T_{inf}$. It can be shown: $f_{filt} \geq 0$. The centroid is now computed on $f_{filt}$. Figure 10(a) shows a portion of a video frame with a leukocyte and the set $S$ (region inside the circular contour). Gradient magnitudes before and after filtering are shown respectively in Figures 10(b) and 10(c).




(a)




(b)

Figure 11. Centroid tracker with and without inclusion filters. (a) Mean position errors. (b) Percentage of frames tracked.

Table 1. Summary of the comparative results.

| | Mean position error (in units of pixel width) | Percentage of frames tracked |
|---|---|---|
| Centroid without inclusion filter | 31 | 13% |
| Centroid with inclusion filter | 11 | 67% |
| Percentage of sequences in which inclusion filter enhances performance | 84 % | 91 % |

We perform 100 tracking experiments on video sequences each of length 91 frames (duration 3 seconds). The value of $R$ is set as 5 pixels. Manually observed leukocyte center

location is utilized to measure the position error for every frame – the distance between centroid tracker computed leukocyte center and manually observed center. A frame is considered successfully tracked if the position error for that frame is less than *R*. The percentage of frames tracked is computed as the ratio of the total number of frames tracked in a sequence over 91. Figure 11 shows the mean position errors and percentages of frames tracked in centroid tracking with and without inclusion filters for the 100 sequences. Table 1 summarizes the comparative results by taking the mean measures over all the 100 sequences. Table 1 shows that the use of inclusion filter increases the number of frames tracked by a factor of 5 and decreases the mean position error by a factor of 3. Note also that out of 100 sequences the mean position error decreases in 84 sequences and the number of frames tracked increases in 91 sequences with the inclusion filter.

## VII. CONCLUSIONS AND FUTURE WORK

In summary, this paper defines a class of connected operator that is perfectly self-dual, increasing and idempotent. We have shown that the proposed filter falls into a larger class of connected operators, known as levelings. Several applications of inclusion filters have been presented here.

In our future work we desire to address a true grayscale description and subsequent fast implementation of the inclusion filter. An example is shown in Figure 12. The top left of Figure 12 shows a grayscale image with four gray levels. The constituent flat zones are labeled as 0, $1_A$,$1_B$, $2_A$,$2_B$, and 3. The corresponding adjacency graph for the flat zones is shown in the top-middle image. The middle and the third row respectively show three level sets of the grayscale image, and the inclusion filtered level sets, where the filter fills small holes. The fourth row image is the output of the inclusion filter. Note that it is possible to convert the adjacency graph into an inclusion tree

[15] as shown on the top right corner of the Figure 12. Then, the same filtering output is obtained by merging the two leaf nodes of this tree with their respective parents. Although fast construction of an inclusion tree is possible via the FLLT (fast level line transform) proposed by Monasse and Guichard [15], the method is not suitable for retaining self-duality (as already shown in Section II). The inclusion structure of the level lines is revealed in the FLLT through the use of upper and lower connected sets. An upper connected set is a set of points where the gray values are greater or equal to a threshold, and a lower connected set comprises of points where the gray values are less than or equal to a threshold. Since these two connected sets cannot simultaneously have 8-adjacency [15], the FLLT cannot be utilized to implement the proposed inclusion filter.

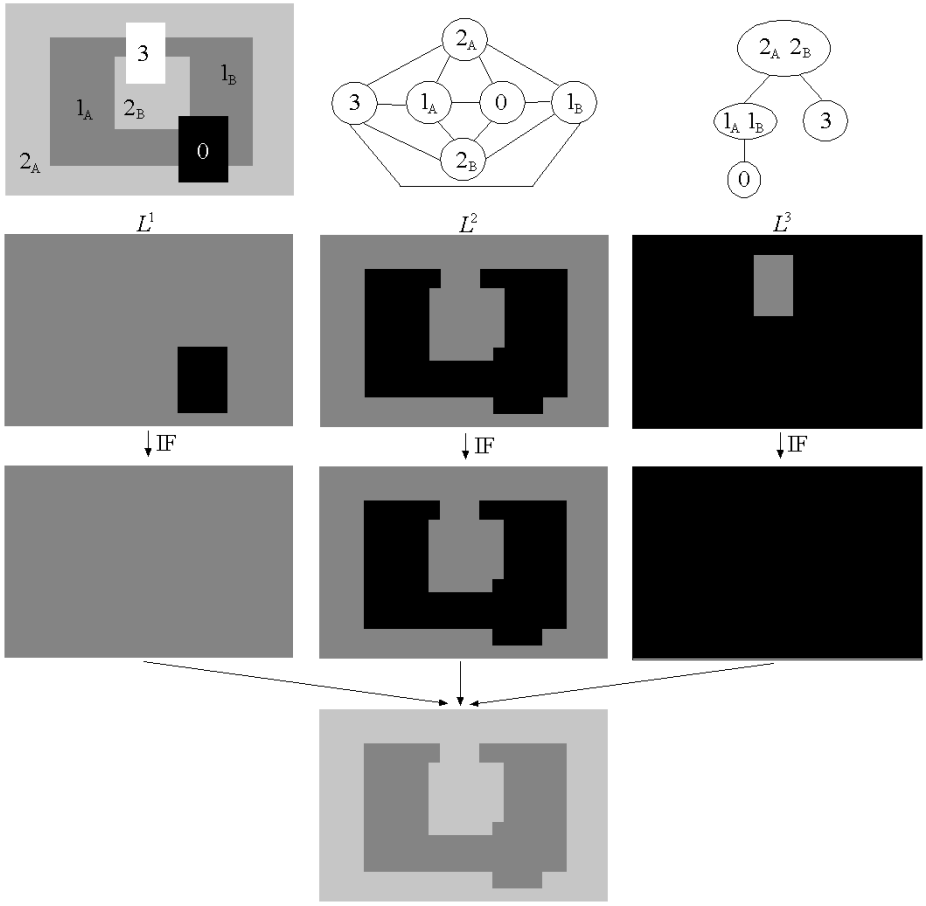



Figure 12. Grayscale image, flatzone adjacency graph, a grayscale inclusion tree, and inclusion filter.

## APPENDIX

We show that if 8-adjacency is considered for both the background and the foreground, then an adjacency tree exists. We define a straightforward algorithm to convert a 4-8 adjacency tree (*i.e.*, 4-adjacency for foreground and 8-adjacency for background) to an 8-8 adjacency tree:

S1: Add a border to the binary image. The added border points then belong to the background set (white pixels). Now construct a 4-8 adjacency tree. The tree will have a single white node, *R*.

S2: Repeat:

for each white node, *W*, in the adjacency tree, do

for each black child node, *B* of *W*, do

if *B* is 8-adjacent to $B_1$, another child of *W*, then

Delete the link between *B* and *W*. Join the sub-tree rooted at *B* to the main adjacency tree by combining *B* with $B_1$.

end

end

end

Until, no further change occurs in the adjacency tree.

S3: Repeat:

for each white node, *W*, in the adjacency tree, do

for each black child node, *B*, of *W* do

if *B* is 8-adjacent to *P*, the parent of W, then

Delete the link between *W* and *B*. Join the sub-tree rooted at *B* to the main adjacency tree by combining *B* with *P*.

end

end

end

Until no further change occurs in the adjacency tree.

S4: for each black child node $B$, of the root, $R$, do

if $B$ is adjacent to the added border in S1, then

Delete the link between $R$ and $B$.

end

end

S5: Let $B_1$, $B_2$,…,$B_n$ are the black children nodes of the root, $R$, that are still linked to it. Delete the links between $R$ and $B_1$, $B_2$,…,$B_n$. Decompose $R$ into $m$ 8-connected components, $R_1$, $R_2$,…,$R_m$, after deleting from $R$ the boundary added in Step 1. Reassign $R_1$, $R_2$,…,$R_m$ as the parents of $B_1$, $B_2$,…,$B_n$.

It is straightforward to show that we obtain an adjacency forest (collection of trees) after the execution of the algorithm. At S1, the algorithm converts the adjacency forest into a tree. S2 and S3 of the algorithm have two basic successive operations. First, a node is deleted from its parent, leaving two trees, and secondly, the trees are combined by joining the two root nodes. Thus after these two successive operations the adjacency tree still remains a tree. The algorithm examines all possible 8-adjacencies among the black nodes; therefore, the tree we obtain at the end of S3 is indeed an adjacency tree with 8-adjacency for both background and foreground. At S4, the foreground root nodes are identified and the tree becomes a forest. To prove that the graph we obtain after S5 is an adjacency forest, we merely need to check that each $B_i$ is assigned a unique parent $R_k$. To show this, we decompose each 8-connected $R_k$ into 4-connected components, $R_k^1, R_k^2, \ldots, R_k^{k_p}$. So, we now must have exactly one 4-connected background component, $R_k^j$, surrounding the 8-connected foreground component $B_i$ [19]. Therefore there is exactly one $R_k$, which is assigned as the parent of $B_i$.

## REFERENCES


[1] S.T. Acton and D.P. Mukherjee, "Scale space classification using area morphology," *IEEE Trans. Image Proc.*, vol. 9, pp. 623-635, 2000.

[2] U.M. Braga Neto and J. Goutsias, "Connectivity on complete lattices: new results," Computer Vision and Understanding, vol. 85, pp. 22-53, 2002.

[3] E.J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Computer Vision and Image Understanding*, vol. 64, pp. 377-389, 1996.

[4] J.F. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 679-698, 1986.

[5] V. Caselles, J.-L. Lisani, J.-M. Morel and G. Sapiro, "Shape preserving local histogram modification," *IEEE Trans*. *Image Proc.*, vol. 8, pp. 220 –230, 1999.

[6] V. Caselles and P. Monasse, "Grain filters," Kluwer Academic publishers, 2001, preprint.

[7] F. Cheng and A.N. Venetsanopoulos, "An adaptive morphological filter for image processing," *IEEE Trans*. *Image Proc.*, vol. 1, pp. 533-539, 1992.

[8] H.J.A.M. Heijmans, "Introduction to connected operators," In *Nonlinear filters for image processing*, E.R. Dougherry and J.T. Astola, Ed., SPIE/IEEE Series on Imaging Sc. and Eng., Piscataway, NJ, 1999.

[9] E. Halimskii (E. Khalimsky), *Uporiadochenie topologicheskie prostranstva* (Ordered Tological Spaces), Naukova Dumka, Kiev, 1977.

[10] T.Y. Kong and A. Rosenfeld, "Digital Topology: Introduction and survey," *Comp*. *Vis*. *Graph*. and *Image Proc*., vol. 48, pp. 357-393, 1989.

[11] S. Masnou, "Image restoration involving connectedness," In *Proc*. *6th international workshop on digital image processing and computer graphics*: *Applications in Humanities and Natural Sciences*, Vienna, Austria, October 1997.

[12] S. Masnou and J.-M. Morel, "Level lines based disocclusion," In *Proc*. *IEEE ICIP*, vol. 3, pp. 259 –263, 1998.

[13] F. Meyer, "Alpha-beta Flat Zones, Levelings and Flattenings," In the *Proceedings of the 6th international symposium on mathematical morphology*, pp. 47-68, 2002.

[14] P. Monasse, "Contrast invariant registration of images," In *Proc*. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. 3221 –3224, 1999.

[15] P. Monasse and F. Guichard, “Fast computation of a contrast-invariant image representation,” *IEEE Trans*. *Image Processing*, vol. 9, pp. 860-872, 2000.

[16] D.P. Mukherjee and S.T. Acton, “Document page segmentation using multiscale clustering,” In *Proc*. *IEEE ICIP*, vol. 1, pp.234-238, 1999.

[17] N. Ray, S.T. Acton, T. Altes, E.E. de Lange and J.R. Brookeman, “Merging parametric active contours within homogeneous image regions for MRI-based lung segmentation,” *IEEE Trans. Medical Imaging*, vol.22, no. 1, 2003.

[18] N. Ray, S.T. Acton, and K. Ley, “Tracking leukocytes *in vivo* with shape and size constrained active contours,” *IEEE Trans. Med. Imag.*, vol.21, no.10, pp.1222-1235, 2002.

[19] A. Rosenfeld, *Picture languages*, Academic Press, New York, NY, 1979.

[20] P. Salembier, A. Olveras, and L. Garrido, “Anti-extensive connected operators for image and sequence processing,” *IEEE Trans*. *Image Processing*, vol. 7, pp. 555-570, 1998.

[21] P. Salembier and J. Serra, “Flat zones filtering, connected operators, and filters by reconstruction,” *IEEE Trans*. *Image Processing*, vol. 4, pp. 1153-1160, 1995.

[22] J. Serra, "Connectivity on complete lattices," *J. Math. Imaging Vision*, vol. 9, pp. 231-251, 1998.

[23] P. Soille, *Morphological image analysis*, Springer-Verlag, Berlin Heidelberg, 1999.

[24] L. Vincent, “Grayscale area openings and closings, their efficient implementation and applications,” In *Proc. First Workshop Mathematical Morphology and its Applications to Signal Processing*, pp. 22-27, Barcelona, Spain, May 1993.

[25] M.H.F. Wilkinson and M.A. Westenberg, “Shape preserving filament enhancement filtering,” In *Medical Image Computing and Computer-Assisted Intervention*, vol. 2208 of *Lecture Notes in Computer Science*, W.J. Niessen and M.A. Viergever, Ed., pp. 770-777, 2001.