Published in final edited form as:
J Electron Imaging. 2012 June 22; 21(2): . doi:10.1117/1.JEI.21.2.023020.

Model-controlled flooding with applications to image reconstruction and segmentation

Quanli Wang and **Mike West**
Duke University, Department of Statistical Science, Durham, North Carolina 27708-0251

Quanli Wang: quanli@stat.duke.edu

Abstract

We discuss improved image reconstruction and segmentation in a framework we term model-controlled flooding (MCF). This extends the watershed transform for segmentation by allowing the integration of a priori information about image objects into flooding simulation processes. Modeling the initial seeding, region growing, and stopping rules of the watershed flooding process allows users to customize the simulation with user-defined or default model functions incorporating prior information. It also extends a more general class of transforms based on connected attribute filters by allowing the modification of connected components of a grayscale image, thus providing more flexibility in image reconstruction. MCF reconstruction defines images with desirable features for further segmentation using existing methods and can lead to substantial improvements. We demonstrate the MCF framework using a size transform that extends grayscale area opening and attribute thickening/thinning, and give examples from several areas: concealed object detection, speckle counting in biological single cell studies, and analyses of benchmark microscopic image data sets. MCF achieves benchmark error rates well below those reported in the recent literature and in comparison with other algorithms, while being easily adapted to new imaging contexts.

1 Introduction

Image segmentation is one of the most central, yet often challenging problems in many areas including biological single-cell studies and computer vision. For example, accurate segmentation of cell nuclei from microscopic images is usually a first and crucial step in single-cell studies in systems biology.[1–3] However, adapting existing algorithms to such problems has proven to be difficult due to complicated imaging conditions, the diversity of image features represented across differing cell types, weak intensity contrast, and touching nuclei, among other issues.[4] Examples in computer vision involve automated detection and segmentation of concealed objects in terahertz images. In this context, specialized algorithms[5] have been developed as general, state-of-the-art algorithms (e.g., level sets[6] and the normalized cut algorithm[7]) typically fail due to weak intensities and low signal-to-noise ratios, coupled with the lack of a frame-work for adapting these approaches to each new, specific population of objects under consideration.

Multiple approaches to segmentation are surveyed in Refs. 8–11. Among advanced segmentation approaches, watershed[12–14] and connected operator methods[10,11,15–17] are among the most generally effective. The watershed transform operates on an input image considered as a topographical surface. Initially implemented in a discrete format via flooding simulations[13] or hierarchical queues,[14] watershed was later recapitulated in a continuous framework using eikonal partial differential equations.[18] Connected operators (e.g., filters by reconstruction, attribute filters) also operate on topographical surfaces but take a quite different approach in filtering connected components at each surface contour. A key difference between these two approaches is that, while they each partition an image into connected components through region growing, the watershed algorithm does this by growing into neighboring pixels sequentially and the connected operator by growing into neighboring regions based on connectivity. Bertrand et al.[19] bridged this gap with the topological watershed transform, defining watershed based on connected operators. This approach puts the connected operator at the heart of watershed transform analysis and enables implementation of the topological watershed via component-tree based algorithms that work very well for connected operators.

Connected operators play a major role in many image reconstruction and segmentation algorithms and continue to be the core of recent active studies.[20–22] Our work builds on this but extends the framework to also admit algorithms that are not necessarily based on connected operators and that can yield improved segmentation results. The innovation here is closely related to a particular implementation of the marker-controlled watershed algorithm, so we provide a brief introduction and its formulation as follows.

Any grayscale image can be considered as a topographic surface or landscape. The original watershed transform is a segmentation method that simulates a landscape being flooded by rain. Rain falling on the surface of a landscape will flow to some regional minima, with all rain falling on a given catchment basin flowing toward the same regional minimum. Watershed lines that separate catchment basins are treated as boundaries between regions for the purpose of segmentation. An alternative model is based on flooding simulation. If the surface is flooded from its regional minima and the merging of water from different sources is prevented by building dams, the image is partitioned into two different sets: catchment basins and watershed lines (dams), defining a natural segmentation of the original image.

Vincent and Soille[13] proposed an efficient algorithmic implementation based on flooding simulations. Formally, let I be a grayscale image under study, with $h_{\min}$ and $h_{\max}$ being the smallest and largest values taken by I in its domain D_I. Let $T_h(I)$ stand for the threshold set function of I at level h:

$$T_h=\{p \in D_I | I(p) \le h\}, \quad h_{\min} \le h \le h_{\max}. \quad (1)$$

The following recursion between gray levels defines the watershed algorithm based on flooding simulation:

$$\begin{cases} X_{h_{\min}}=\{p \in D_I | I(p)=h_{\min}\}, \\ X_{h+1}=\mathrm{MIN}_{h+1} \cup \mathrm{IZ}_{T_{h+1}(I)}(X_h), \quad h_{\min} \le h<h_{\max}, \end{cases} \quad (2)$$

with X_h being the union of the set of catchment basins computed at level h, MIN_h the union of all regional minima at level h, and $\mathrm{IZ}_A(B)$ the union of geodesic influence zones of the connected components of B with $B \subseteq A$. Further details on $\mathrm{IZ}_A(B)$ can be found in Ref. 13. In this approach, the algorithm starts from the lowest regional minima $X_{h_{\min}}$ at level $h_{\min}$. $X_{h_{\min}}$ is then flooded at level $h_{\min+1}$ represented by $T_{\min+1}$ The geodesic influence zone function IZ is then employed to propagate $X_{h_{\min}}$ to $X_{h_{\min+1}}$ based on geodesic distance while preventing merging of regional minima by building watershed lines. This process continues until the highest regional maxima are reached at level $h_{\max.}$

In general, the watershed transform leads to severe over-segmentation in contexts of high frequency noise in the input image; this is usually dealt with by preprocessing steps using techniques such as filters by reconstruction or attribute filters. This issue is intrinsic as segmentation is completely driven by the regional minima, *regardless of other prior information about the objects of interest*. Marker-controlled and hierarchical watershed methods were developed as remedies by either providing strong prior information ("markers") to preprocess the image before applying watershed transforms,[14] or via merging rules to postprocess oversegmented regions.[23–27] The success of these methods depend heavily on the choice of initial markers and the rules specific to the image, and it has proven difficult to define generally useful methods.[27,28] In an attempt to include both intensity contrast and region size criteria into the flooding process, a generalized method was introduced in Ref. 29 for the partial differential equation formulation. The resulting uniform volume flooding method showed improvement for certain problems but adapting this method to different image characteristics is a complex task. Vargas-Vazquez et al.[30] recently introduced an approach based on the watershed transform, which skips the filtering step by adding a shape-related criterion to the flooding process to control how image structures are recovered.

Despite active research in the field, image segmentation remains a challenging problem due to the diversity and complexity of application contexts. While connected operators can be parameterized in the form of attribute filters, the choices for attributes satisfying connected operator constraint are limited. This makes generality and portability of such methods a major challenge. Use of the watershed algorithm incorporating criterion suffers from the same issue. Using the geodesic influence zone function as the only propagating model in the flooding process makes it extremely hard to generalize to other attributes.

Our work here defines a general, adaptable approach to addressing all these issues. The novel model-controlled flood (MCF) simulation framework represents advances toward effective and automatic use of context-sensitive information in the flooding process; this generalizes both the watershed algorithm and connected operators by introducing modeling and parameters to make the process more general and portable. MCF allows users to define application-specific, parameterized, plug-in functions that utilize prior information specific to the problem to control how the objects of interest should behave during a recursive flooding simulation process. The specification is direct and intuitive, as our examples demonstrate, and the results can represent substantial improvements over existing approaches. This paper provides detailed descriptions and illustrations of the following aspects of the MCF algorithm:

- Description of the novel model-controlled flooding (MCF) framework for image reconstruction and segmentation;
- Definition of the MCF-based size transform and its applications to binary and gray scale image segmentation, and discussion of advantages of the framework over attribute filters and watershed algorithms;
- Examples of biological cell counting using customized size transforms on benchmark human and fly image collections;
- Examples of using an MCF-based algorithm exploiting both size and shape information for detection and segmentation of concealed objects from terahertz images;
- Examples of MCF size/shape transforms for single molecular identification, or so-called "speckle counting," in biological single cell studies.

The novel MCF framework is introduced in Sec. 2, and its connections with—and distinctions from—the marker-controlled watershed algorithm and attribute filters are discussed. Section 2 also introduces a special case of the MCF implementation—the size transform—followed by some representative segmentation examples to demonstrate its simplicity and superior performance relative to existing algorithms. Section 3 discusses size transform customization using shape information, aspects of performance evaluation, parallel implementations, and a variety of examples to further justify the advantages of the proposed framework. The MCF algorithm performance is further validated using a total of 56 benchmark microscopic image sets from the Broad Institute Bioimage Benchmark Collection.[31] With benchmark error rates well below published results, the proposed algorithm substantially improves segmentation results across multiple examples, while being general, easily customizable and portable. Section 4 further demonstrates the portability of the algorithm by applying and adapting the algorithm to two very different, topical applications: detection and segmentation of concealed objects without modification, and speckle counting in single cell studies with minor customization. Concluding comments appear in Sec. 5.

2 MCF Framework and Size Transform

2.1 MCF Framework

Let $T_h(I)$ be the threshold set of an image I at level h as defined in Eq. (1). We define the model-controlled flooding (MCF) framework by the following recursion between gray levels of the image:

$$\begin{cases} X_{h_{\min}} = \mathrm{CF}(\varnothing, T_{h_{\min}}(I), \theta), \\ A_{h_{\min}} = X_{h_{\min}}, \\ X_{h+1} = \mathrm{CF}(X_h, T_{h+1}(I), \theta), \quad h_{\min} \le h < h_{\max}, \\ A_{h+1} = \mathrm{AGR}(A_h, X_{h+1}, \phi), \quad h_{\min} \le h < h_{\max}, \end{cases} \tag{3}$$

where $h_{\min}$, $h_{\max}$, X_h are as in Eq. (2), $\mathrm{CF}(B_1, B_2, \theta)$ is any function that maps binary images B_1 and B_2 into another binary image of the same size using a given parameter θ, A_{h+1} is a grayscale image and $C = \mathrm{AGR}(A, B, \varphi)$ is any function that maps a grayscale image A and a binary image B into another grayscale image C using a given parameter vector ϕ subject to

$$\forall\, p \in D_I, \quad A(p) > 0 \Rightarrow C(p) > 0. \tag{4}$$

Compared to Eq. (1) for watershed algorithm, the MCF framework described by Eq. (3) is novel in three aspects. First, it introduces a second status variable A_h to maintain a grayscale image at each flooding level h, which makes it potentially also an algorithm for reconstruction. Second, it allows the use of parameters at each step. These parameters are not associated with any particular feature of an image in the framework but are provided as a way to incorporate prior information regarding the specific segmentation/reconstruction problem into the flooding process. Examples will be given in later sections on the use of such parameters. Third, two functions CF, for controlled flooding, and AGR, for aggregation, are introduced to update X_h and A_h respectively. Instead of hardwiring a geodesic influence zone function into the flooding process as in Eq. (1) for updating X_h, the framework makes it more flexible by allowing user-defined, plug-in functions in the updating process. Similar to the role of the geodesic influence zone function in the watershed algorithm, CF acts as an object modeling tool to decide how the result from the previous level h should update/expand itself based on the newly available information T_{h+1}, the threshold set at current level $h+1$. The newly introduced function AGR then acts as a moderator to decide how to combine the cumulative result A_h from previous levels with the newly updated result X_{h+1} at the current level, possibly using prior information in the parameter vector φ. Equation (4) is introduced merely to impose restrictions on CF and AGR such that flooding at a higher level does not invalidate the previous result.

The MCF framework described by Eq. (3) is illustrated in the flow chart of Fig. 1. Suppose that the goal is to identify the round-shaped object from the contaminated input image. The input image is first inverted and viewed as a topographic surface. The lowest level $h_{\min}$ is identified to initialize two status variables as shown in the second row; they are trivial in this case. As the flooding level moves up (row 3 to row 7), the threshold set is first identified (the red arrow). Then a controlled flooding function CF (the green arrows) is deployed to extract information of interest using the result from the previous step and the new thresholding set. Some straightforward information about this object is that it is round, solid, and its size in terms of pixels is within a certain range. Taking this prior information as parameters, the CF function can score the objects in the threshold set in column 1 and take actions as necessary. For example, it removes the objects in row 3 that are not round or that are too small; it can fill the hole in the object in row 4 that has the right size but is not solid; it can select the round object in row 5 but discard the elongated blob from the same threshold set; it can further remove objects in row 6 that are either too big or too small. The results of applying CF function are shown in column 2 and only the objects that fit the model are preserved and/or enhanced. The AGR function (the orange arrows) is shown in Fig. 1 as column 3, which in this case simply stacks up all the results from column 2 without using any parameters. As suggested by the final result $A_{h_{\max}}$ in row 7, column 3, and its topographic view in the bottom row, the reconstructed image using MCF is much cleaner than the original image.

The final result from Eq. (3) can be in the form of a +reconstructed grayscale image to be further segmented using existing methods, or in terms of just a binary mask that represents the final segmentation result, depending on how the aggregation function is formed. We will, however, demonstrate that the reconstructed images usually have very desirable features that can be easily segmented using standard existing methods.

2.2 MCF Examples

Based on Eqs. (3) and (4) we provide two MCF examples using different CF and AGR functions. Our first example demonstrates that the marker-controlled watershed algorithm is a special case of MCF. The second example provides a simple MCF size transform and is used in the remaining sections to illustrate the approach.

Marker-controlled watershed algorithms (with criterion)—Let AGR $(A, B, \phi) = B$ and CF $(B_1, B_2, \theta) = IZ_{B_2}(B_1) \cup [B_2 \setminus R(B_2, B_1)]$, where $R(B_2, B_1)$ is the opening by reconstruction operator.[32] Then plugging AGR and CF into Eq. (3) yields

$$\begin{aligned} A_{h_{\min}} &= X_{h_{\min}} = \mathrm{CF}[\varnothing, T_{h_{\min}}(I), \theta] \\ &= \mathrm{IZ}_{T_{h_{\min}}(I)}(\varnothing) \cup \{T_{h_{\min}}(I) \setminus R[\varnothing, T_{h_{\min}}(I)]\} \\ &= \varnothing \cup T_{h_{\min}}(I) = T_{h_{\min}}(I) \end{aligned}$$

and

$$\begin{aligned} A_{h+1} &= \mathrm{AGR}(A_h, X_{h+1}, \phi) = X_{h+1} = \mathrm{CF}[X_h, T_{h+1}(I), \theta] \\ &= \mathrm{IZ}_{T_{h+1}(I)}(X_h) \cup \{T_{h+1}(I) \setminus R[X_h, T_{h+1}(I)]\} \\ &= \mathrm{IZ}_{T_{h+1}(I)}(A_h) \cup \{T_{h+1}(I) \setminus R[A_h, T_{h+1}(I)]\} \\ &= \mathrm{IZ}_{T_{h+1}(I)}(A_h) \cup \mathrm{MIN}_{h+1}. \end{aligned}$$

Note that the last step is obtained by observing the fact that the regional minima MIN_{h+1} are the points that belong to threshold set T_{h+1} but not $R[A_h, T_{h+1}(I)]$. This makes the marker-controlled watershed transform a special case of MCF. It is then obvious that the watershed transform does not take any prior information into consideration during its flooding process. Also, region growing is wholly defined by the influence zone function. However, a new variant of marker-controlled watershed was recently introduced to include a criterion or shape information into the flooding process.[30] This modifies the threshold set at each flooding level through mathematical opening with parameterized structure elements before computing influence zones. If we set $\mathrm{CF}\,(B_1, B_2, \theta) = \mathrm{IZ}_{\gamma_\theta(B_2)}\,(B_1)$, where γ_θ is the mathematical opening operator with structure element size θ, our controlled flooding function then describes the algorithm proposed in Ref. 30, making it a parameterized special case of MCF.

Size transform—In many image segmentation problems, the sizes of objects in a given image are usually the most recognizable features other than intensity values. The following realization of MCF gives a very simple example that exploits this feature.

Suppose B_2 is a binary image of k connected components $Z_1, Z_2,\ldots,Z_k$ with corresponding areas (numbers of pixels) $P_1, P_2,\ldots,P_k$. Define $\theta = (\mathrm{size}_{\min}, \mathrm{size}_{\max})$ by two prior values $\mathrm{size}_{\min}$, and $\mathrm{size}_{\max}$, and set φ to be a vector of zero length. Further, set

$$\begin{cases} \mathrm{CF}(B_1, B_2, \theta) = \bigcup\limits_{\{t \mid \mathrm{size}_{\min} \le P_t < \mathrm{size}_{\max}\}} Z_t, \\ \mathrm{AGR}(A, B, \varphi) = A + B, \end{cases} \tag{5}$$

in the MCF algorithm. This specific CF function simply filters out from the threshold set B_2 any connected components that are too small or too large; AGR simply aggregates the result by adding up the filtered binary masks for the threshold sets at all levels. We call this realization in Eq. (5) an MCF size transform as it only uses the minimum and maximum sizes for object modeling.

Compared to the watershed algorithm, this formulation is very simple but illustrates the very fundamental aspect of the framework: using simple functions to integrate prior information into the flooding process. The minimum size parameter $\mathrm{size}_{\min}$ here provides an automatic way of finding initial seeds or markers, while the maximum size parameter $\mathrm{size}_{\max}$ provides a stopping rule for the flooding process. Region growing in the process is simplified by accumulation of binary masks from each flooding level and is guaranteed by Eq. (4). Further, it is clear that the size transform is not an attribute filter as defined in Ref. 16 and that the grayscale area opening is a special case of this transform if only the minimum size parameter is used. Attribute filters are strictly built on connected operators and changes to the threshold sets are made only through connected components; in contrast, MCF allows any changes to be made on each pixel based on a model function, as shown via the customized size transforms in following sections. In other words, the MCF framework provides extension of both attribute filter and watershed transform approaches. It allows region growing based on a model and prior parameters in a flooding process. Importantly, it is not necessarily hardwired to a connected operator and attribute as in the case of attribute filters, nor is it hardwired to a specific function—such as the influence zone function—as in the watershed transform.

2.3 Representative Segmentation Examples Using Size Transform

We present three examples that involve directly applying the MCF size transform as defined in Eq. (5).

Example 1. Identifying pears in a gray scale image—Figure 2(a) gives the grayscale input image of several pears from the Matlab Image Processing Toolbox. The goal is to identify/separate pears from the background and from each other. Figure 2(b) is the typical resulting mosaic image after applying the watershed transform to the gradient magnitude of Fig. 2(a). It is evident that the image is highly over segmented due to the blemishes and high frequency noise. Figure 2(c) is obtained using the marker-controlled watershed transform as suggested in Ref. 3. The foreground markers are generated by a multistep pipeline involving edge detection, image smoothing using "filters by reconstruction," and global thresholding. The background markers are calculated by applying watershed transform to the distance map of the Otsu thresholding[33] of the original image Fig. 2(a).

Figure 2(d) is the result $A_{h_{\max}} > 0$ after applying the proposed MCF size transform Eq. (5) based on the parameter $\theta = (500, 17500)$, i.e., we assume that all the pears to be identified will have a size of at least 500 pixels and at most 17,500 pixels, rough visual estimates of the pear sizes in the image. Figure 2(c) and 2(d) uses random colors to indicate the identified pears and the segmentation results are superimposed on the original image. This simple MCF size transform clearly identifies more pears correctly in a coherent, naturally and easily understood way.

Example 2. Segmentation of a binary particle image—This example was analyzed in Ref. 34 using a six-step adaptive watershed algorithm to segment irregular-shaped binary images based on the concept of overlap parameter. The simple procedure we use is implemented in three steps:

- Step 1: Calculate the distance map of the input binary image.
- Step 2: Apply the MCF size transform to the distance map and extract local maxima as markers. Set $\text{size}_{\max}$ to a very large value to just filter out oversized objects.
- Step 3: Segment the distance map from step 1 according to the markers from step 2, using the marker-controlled watershed algorithm.

While the first and the third steps are standard for most binary segmentation problems, the second step is the most important: it uses the MCF size transform to reduce the spurious local maxima to achieve satisfactory results; see Fig. 3.

Figure 3(a) is the input irregular-shaped binary particle image. Figure 3(b) shows the result from the conventional watershed algorithm using the Matlab watershed transform implementation; this is a special case of MCF size transform with $\text{size}_{\min} = 0$. Over-segmentation is clearly evident. Figure 3(c) and 3(d) gives results using the size transform with $\text{size}_{\min} = 25$ and $\text{size}_{\min} = 100$. Figure 3(e) plots the total number of segmented particles against the single nontrivial parameter $\text{size}_{\min}$. Further investigation shows that the proposed algorithm: a) produces oversegmented results when $\text{size}_{\min}$ is smaller than 7; b) produces under-segmentation results when $\text{size}_{\min}$ is larger than 157; c) gives the same result as shown in Fig. 3(c) for any parameter between 8 and 52; d) gives the same result as shown in Fig. 3(d) for any parameter between 53 and 157. It is evident that the algorithm is relatively insensitive to the choice of parameter and produces very stable and satisfying results across a wide range of values of this single parameter. Testing on multiple other binary images suggests that a default setting of $\text{size}_{\min} = 10$ usually suppresses most spurious local minima and yields good segmentation results.

Example 3. Cell segmentation using microscopic images of budding yeast— Cell segmentation of clustered budding yeast from a microscopic image is a critical challenge for many biological studies.[35] Here we use the size transform to both identify the foreground and segment the foreground into individual cells for such images. A three-step procedure is used:

- Step 1: Apply the size transform to reconstruct the images to enhance the contrast between the foreground (regions of cell clusters) and the background.
- Step 2: Identify the foreground by thresholding the result from step 1.
- Step 3: Use the procedure described in example 2 to segment the binary foreground for individual budding yeast cells.

Figure 4(a) shows the input testing microscopic image of budding yeast colonies, generally shown as the dark and round, tightly clustered regions. Figure 4(b) gives the result $A_{h_{\max}}$ after step 1, using the trivial parameter $\theta = (1, a/4)$, where a is the total number of pixels in the image. Figure 4(c) shows the extracted foreground after step 2; this suppresses those maxima of Fig. 4(b) with height less than 20; this uses the Matlab function ihmax, and any number between 10 and 30 will give very similar results. Figure 4(d) gives the final segmentation result after step 3 and using the default parameter setting.

A very simple global thresholding method converts the transformed result in step 1 into a binary image in step 2. In case of dealing with a series of similar images, other methods such as Local Histogram Equalization or Rank Transform on Hybrid images[3] can be employed to avoid manually choosing a global threshold.

3 Discussions and Comparisons

The size transform Eq. (5) demonstrates the ability to incorporate prior information into the flooding simulation process to shape the final results. By customizing the CF and/or AGR functions according to the unique features of a problem we can apply this approach to a broad range of imaging studies. For example, for most biological cell nuclei segmentation problems we can assume that the nuclei are convex with somewhat smooth borders. This can be reflected in the MCF function by applying morphological opening/closing to the threshold set for border smoothing and by “filling holes” before or after filtering out unwanted objects based on size. The resulting CF function no longer operates on the connected components and thus does not fall into the category of attribute filters. We give examples using this modification in the following section, analyzing 56 images from two benchmark collections from the Broad Bioimage Benchmark Collection (BBBC).[36] In these examples we calculate benchmarks and make comparisons with two leading segmentation algorithms: CellProfiler[37] and Gradient Flow Tracking.[38]

3.1 Benchmark Image Sets for Cell Counting

To test and evaluate existing or new algorithms for bioimage analysis, we refer to the Broad Institute at MIT collection of freely downloadable microscopic images and “ground truth” (expected results) obtained by human experts. The BBBC site also defines benchmarks for matching ground truth in cell counting, segmentation, and so forth. We use two collections on cell nuclei counting from microscopic images of Human HT29 Colon Cancer 1 cells (human) and Drosophila KC 167 1 cells (fly), as have published benchmark results from CellProfiler. Further, on these data sets the historically large variation in “ground truth” assessments from human observers indicate that these are nontrivial and challenging test images. More details about these data sets can be found at Ref. 36; we note that there are a total of 50 fly image data sets and six cancer image data sets in these collections, and the performance results reported come from the full set of 56 example analyses.

We summarize the results of our comparative analyses in Tables 1 and 2 and Fig. 5. Table 1 lists details on the data sets, methods, parameters, and benchmark calculations. The same MCF size transform is applied to both data sets with different parameters. Table 2 gives results compared to the published results from CellProfiler and Gradient Flow Tracking.

MCF outperforms the published CellProfiler results, improving benchmarks from 17% and 6.7% to 3.99% and 0.1% respectively. While Gradient Flow Tracking performs almost as well as MCF for the human data set, it performs poorly for the fly data set. Figure 5 gives segmentation results for randomly selected images from both data sets using different methods. The first row in Fig. 5 shows input images from the human and two fly samples; the second row gives CellProfiler results, the third row those from Gradient Flow Tracking, and the final row from MCF. It is evident that MCF size transform produces results that are consistently and significantly less over- or undersegmented.

3.2 Parallel Implementation

In general, as we drop the connected operator requirement from the MCF framework, we will also abandon component-tree based approaches[17] that dominate the attribute filter algorithms. However, by modeling threshold sets at each level separately, the MCF size transform and its variants are immediately suitable for parallel implementation. In contrast to the marker-controlled watershed algorithm, the model function described in Eq. (5) does not depend on the first input B_1, or X_h in Eq. (3). Hence, we can first calculate X_h independently and in parallel at each level h. Then all results at all levels are summarized based on the AGR function. The recursive algorithm in Eq. (3), as illustrated by the flowchart in Fig. 6, then naturally fits into the popular Map/Reduce parallel algorithm scenario[39] by treating CF as a Map step and AGR as a Reduce step. As multicore computers[40] and Graphics Processing Unit (GPU) becomes more widely used, the potential for modeling threshold sets independently in parallel are immense and should be exploited.

3.3 Sensitivity to Parameters

The MCF analysis for the benchmark data sets depends on three essential but typically easily assessable parameters:

- maximum “object” size,
- minimum disk structure element radius chosen to smooth object borders,
- minimum highest intensity value that an object has to reach.

The maximum object size is usually determined by visually identifying and estimating the largest object in the image, and generally has a negligible influence on the final segmentation/counting result. The minimum disk structure element is, in contrast, often critical to the final segmentation result and generally serves as a tuning parameter. The minimum highest intensity value is introduced mainly to filter out the objects that are too dim to be considered and usually can be chosen with ease. For our biological examples, Fig. 7 gives plots of the benchmark error rates for the human and fly data sets as a function of minimum structure element radius when the other two parameters are fixed. The horizontal lines give the published benchmarks. We see that inappropriate choice of this parameter can cause over-segmentation when it is too small or undersegmentation when it is too large. Also, there is a wide range of parameter values in each case that lead the MCF analysis to outperform published results, confirming its relative parameter insensitivity and robustness.

4 Additional Applications of MCF

We explore two further applications to demonstrate scope and performance of MCF in very different and nontrivial image analysis problems of broad current interest.

4.1 Detection and Segmentation of Concealed Objects in Terahertz Images

Images of objects concealed beneath the clothing of a human subject can be obtained using terahertz imaging techniques, with applications to automatic detection and segmentation of concealed objects such as guns and knives. Standard segmentation methods have as yet proven to deal poorly with these images due to inherent physical properties of the terahertz imaging technique, i.e., poor contrast and low signal to noise ratio. Recent studies of state-of-the-art segmentation methods and a new unsupervised algorithm that outperforms them are given in Ref. 5. This new algorithm takes a three-step approach: (i) reduce noise using a NL-Means image denoising algorithm,[41] (ii) classify pixels from a cleaned image into three "rough" regions (background, human body, and concealed objects) using Gaussian mixture modeling or similar standard methods,[42] and (iii) apply multilevel thresholding to further identify concealed objects from the human boundary that was misclassified in step (ii).

MCF can be readily customized to segment concealed objects; see Fig. 8(a). The strategy is to use size transform to first segment the background from the rest, and then to segment the concealed objects from the human body.

Background detection—The human body and the background in testing images are relatively large connected regions. Successful segmentation is likely if we can smooth both regions separately without blurring the region boundary. We accomplish this using the size transform with the same size parameters twice: first, on the original image to generate a reconstructed image A; second, on the complement of the original image to a reconstructed image B; see red and green channels in Fig. 8(b). Evidently, both background and human regions are cleaner with borders preserved. Since the goal is to remove noise, the choice of size parameters is trivial and results are quite insensitive to broad ranges of values. For example, the min and max sizes are taken here as 10% and 90% of the total number of pixels in the input image, respectively. If we change these to 1% and 99%, we obtain basically the same results. To further identify the background, we simply mark a pixel as background if its value in A is larger than that in B, which effectively segments the background from the rest in each case; see Fig. 8(c).

Concealed object detection—To identify the concealed objects from a terahertz image using the same size transform, we assume that: (i) any concealed object has a minimum size in pixels of at least 1% of the size of the human body, (ii) a maximum size of at most 10% of that of the human body, and (iii) the object boundary can be smoothed by a mathematical morphological opening of size 2. Figure 8(d) gives the results of applying the MCF size transform to Fig. 8(c). This reliably highlights all the concealed objects which can then be easily identified using simple global thresholding as suggested in Sec. 3. The parameters are not optimized in any way, but already show the ability of the simple and automatic MCF analysis. A more rigorous approach would involve training the parameters using a subset of manually labeled images.

4.2 Speckle Counting

As a broad platform for biological imaging, CellProfiler[37] provides an advanced pipeline for Speckle Counting, to identify smaller objects (foci) within larger objects (nuclei) with very many tunable parameters for user customization. Some of the important parameters are size and smoothness of both the larger and smaller objects. We show here that the MCF size transform can be used for all the tasks involved here, including cell nuclei identification, touching cell nuclei separation and foci detection, automatically and with minimal customization.

Nuclei detection—Figure 9(a) gives an example from Cell-Profiler. It seems easy for human eyes to segment, but poses problems when an algorithm is applied due primarily to the weak gradient surrounding cell nuclei and also the large intensity variation within each. We reconstruct the image by smoothing both background and foreground, keeping the cell nuclei boundary relatively unchanged. Figure 9(b) shows the reconstructed result using MCF size transform with the same size parameters as in the previous example (10% of total pixels for minimum size and 90% for maximum size). As expected, both background and cell nuclei are more homogenous in intensity distribution and the borders are preserved. A simple global thresholding such as Otsu's method yields the cell nuclei identification; see Fig. 9(c).

Touching nuclei separation—While Fig. 9(c) gives a binary mask of all nucleus within the image, touching nuclei have to be further separated. This can be done easily using the procedure described in Sec. 2 on segmentation of a binary particle image; see Fig. 9(d). The single size parameter can range from 20 to 5000 without changing the final nuclei segmentation result.

Foci detection—Figure 9(e) shows the image that captures the foci. As suggested in the CellProfiler pipeline, all foci will have minimum and maximum sizes and certain border smoothness. We characterize this prior information by setting the minimum and maximum size parameters to 10 and 100 pixels respectively, and also by assuming that the detected foci will be smoothed by a morphological opening of size 1. Applying MCF size transform to data in Fig. 9(e) yields results as in Fig. 9(f) (log scale); generally, some foci will be touching each other, although their centers are identified by local maximum regions that can be further segmented using a watershed algorithm. A direct approach avoiding this problem is to simply customize MCF via the AGR function in order to only include a component if its inclusion does not invalidate the maximum size restriction defined in the model function. This results in Fig. 9(f), defining the final estimation of image foci. Figure 9(g) and 9(h) gives close-ups of the upper-right nuclei. For comparison, the corresponding segmentation result for that upper-right region from the CellProfiler pipeline using its default parameters is shown in Fig. 9(i).

5 Conclusions

The Model-Controlled Flooding framework generalizes both the marker-controlled watershed algorithm and attribute filters by allowing users to integrate prior information regarding objects of interest directly into the flooding simulation process and by relaxing the connected operator constraint on modeling functions applied to threshold. Using the size transform and its customized versions as examples, we demonstrate that simple model functions operating on binary images can be effectively used to control all aspects of the flooding process: providing initial flooding regions (initial markers) implicitly through minimum size requirements in a region, deciding when the flooding should stop through maximum size of a region, and determining how a region should grow by using shape and smoothness constraints. The aggregation function adds an extra layer of flexibility by allowing the process to decide how to combine binary results from different threshold set levels. The end results of the proposed algorithm are reconstructed images with improved contrast and more homogenous intensity distributions, images that can be further segmented using existing methods. By contrast, the marker-controlled watershed transform uses only predefined initial markers as starting points, and hardwires the influence zone function as a model function during the flooding process. The attribute filters only operate on the connected regions of threshold sets and do not allow alteration of flat zones.[11]

We have defined a size transform as a special case of the MCF algorithm and used it to effectively segment both irregular-shaped binary particle images and grayscale images. We have further demonstrated this new approach through a variety of reconstruction and segmentation examples. Using a modified size transform that incorporates shape-related information into the model function, we successfully segmented cell nuclei in the microscopic images from two benchmark image sets of human and fly data, and achieved substantial improvements over contemporary, published benchmark error rates. With examples for different image processing problems such as segmentation of concealed objects from terahertz images and speckle counting, we demonstrated that the proposed algorithm can be used to solve a diverse range of practical segmentation problems, with easy and robust customization through simply interpreted, low-dimensional parameters defining model and aggregation functions. A range of numerical experiments indicate that the proposed algorithm is relatively insensitive to the choice of parameters compared to other leading algorithms, and further examples and experiments (not reported) add to the empirical evidence supporting the view that of the approach can substantially improve segmentation accuracy. We also note that the proposed MCF size transform algorithm is intrinsically parallel and can be readily implemented in a parallel computing environment.

However, we also note that the added flexibility of using modeling functions in flooding process can invalidate existing algorithms. These algorithms become efficient by taking advantages of some intrinsic features. The component-tree approaches can no longer in general be employed as they are based on the connected operator constraint. Also, the flooding algorithm for watershed transform cannot be used, as the pixel queue used in the implementation cannot be defined in advance. We propose parallel implementation ideas but further investigations of implementations with particular modeling functions are desirable.

Future developments will enhance the utility of these methods, in areas including automated methods for parameter selection, additional model and aggregation functions suited to other segmentation problems, and integration of MCF into imaging packages such as CellProfiler.

Acknowledgments

The authors are grateful for the constructive comments of the editorial team and two anonymous reviewers on the original version of the paper. This work was supported in part by the U.S. National Institutes of Health under grants P50GM081883 and RC1-AI086032, and the U.S. National Science Foundation under grant DMS-1106516. Any opinions, findings and conclusions, or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of the NIH or NSF.

The software used for all examples and bench-marking studies is available free to interested readers who may wish to reproduce the results and/or apply the methods to other problems. This, together with additional imaging examples, will be available at the authors' website via the online journal page (http://stat.duke.edu/research/software/west/MCF/).

References

1. Rosenfeld N, et al. Gene regulation at the single-cell level. Science. 2005; 307(5717):1962–1965. [PubMed: 15790856]
2. Gordon A, et al. Single-cell quantification of molecules and rates using open-source microscope-based cytometry. Nat Methods. 2007; 4:175–181. [PubMed: 17237792]
3. Wang Q, et al. Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. Cytometry Part A. 2010; 77A:101–110.
4. Megason SG, Fraser SE. Imaging in systems biology. Cell. 2007; 130(5):784–795. [PubMed: 17803903]
5. Shen X, et al. Detection and segmentation of concealed objects in terahertz images. IEEE Trans Image Process. 2008; 17(12):2465–2475. [PubMed: 19004716]
6. Chan TT, Vese LA. Active contours without edges. IEEE Trans Image Process. 2001; 10(2):266–277. [PubMed: 18249617]
7. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell. 2000; 22(8):888–905.

8. Freixenet J, et al. Yet another survey on image segmentation: region and boundary information integration. Lecture Notes Comput Sci. 2002; 2352:21–25.

9. Sezgin M, Sankur B. Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imag. 2004; 13(1):146–168.

10. Salembier P, Ferra J. Flat zones filtering, connected operators, and filters by reconstructions. IEEE Trans Image Process. 1995; 4(8):1153–1160. [PubMed: 18292010]

11. Meijster A, Wilkinson MHF. A comparison of algorithms for connected set openings and closings. IEEE Trans Pattern Anal Mach Intell. 2002; 24(4):484–494.

12. Meyer F, Beucher S. Morphological segmentation. J Vis Commun Image Represent. 1990; 1(1): 21–46.

13. Vincent L, Soille P. Watershed in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal Mach Intell. 1991; 13(6):583–598.

14. Beucher S, Meyer F. The morphological approach to segmentation: the watershed transformation. Mathematical morphology in image processing. Opt Eng. 1993; 34:433–481.

15. Vincent, L. Morphological area openings and closings for grey-scale images. Proc. NATO Shape Picture Workshop; Springer-Verlag, Driebergen, The Netherlands. 1992. p. 197-208.

16. Breen EJ, Jones R. Attribute openings, thinnings and granulometries. J Comput Vis Image Understanding. 1996; 64(3):377–389.

17. Salembier P, Oliveras A, Garrido L. Anti-extensive connected operators for image and sequence processing. IEEE Trans Image Process. 1998; 7:555–570. [PubMed: 18276273]

18. Maragos P, Butt MA. Curve evolution, differential morphology, and distance transforms applied to multiscale and eikonal problems. Fundam Inf. 2000; 41:91–129.

19. Bertrand G. On topological watersheds. J Math Imag Vis. 2005; 22(2–3):217–230.

20. Najman L, Couprie M. Building the component tree in quasilinear time. IEEE Trans Image Process. 2006; 15(11):3531–3539. [PubMed: 17076410]

21. Westenberg MA, Roerdink JBTM, Wilkinson MHF. Volumetric attribute filtering and interactive visualization using the max-tree representation. IEEE Trans Image Process. 2007; 16(12):2943–2952. [PubMed: 18092594]

22. Ouzounis GK, Wilkinson MHF. Hyperconnected attribute filters based on k-flat zones. IEEE Trans Pattern Anal Mach Intell. 2011; 33(2):224–239. [PubMed: 21193806]

23. Beuchar, S. Watershed, hierarchical segmentation and waterfall algorithm. In: Serra, J.; Soille, P., editors. Proc Mathematical Morphology Applications Image Signal Process. Kluwer Ac. Publ; 1994. p. 69-76.

24. Najman L, Malik J. Geodesic saliency of watershed contours and hierarchical segmentation. IEEE Trans Pattern Anal Mach Intell. 1996; 18(12):1163–1173.

25. UmeshAdigam PS, Chaudhuri BB. An efficient method based on watershed and rule-based merging for segmentation of 3-D histopathological images. Pattern Recogn. 2001; 34:1449–1458.

26. Mao KZ, Zhao P, Tan PH. Supervised learning-based cell image segmentation for P53 immunohistochemistry. IEEE Trans Biomed Eng. 2006; 53:1153–1163. [PubMed: 16761842]

27. Lin G, et al. Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei. Cytometry Part A. 2005; 63A:20–33.

28. Chen X, Zhou X, Wong STC. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. IEEE Trans Biomed Eng. 2006; 53(4):762–766. [PubMed: 16602586]

29. Sofou A, Maragos P. Generalized flooding and multicue PDE-based image segmentation. IEEE Trans Image Process. 2008; 17(3):364–376. [PubMed: 18270125]

30. Vargas-Vazquez D, Rios-Moreno J, Trejo-Perea M. Reconstruction with criterion from labeled markers: new approach based on the morphological watershed. J Electron Imag. 2010; 19:043001.

31. Broad Bioimage Benchmark Collection. http://www.broadinstitute.org/bbbc/

32. Vargas-Vazquez D, Terol-Villalobos IR. Openings and closings with reconstruction: a study of a class of lower and upper levelings. J Electron Imag. 2005; 14:013006.

33. Otsu N. A threshold selection method from gray-level histogram. IEEE Trans Syst, Man, Cybern. 1979; 9:62–66.

34. Sun HQ, Luo YJ. Adaptive watershed segmentation of binary particle image. J Microsc. 2009; 233(2):326–330. [PubMed: 19220699]

35. Elowitz MB, et al. Stochastic gene expression in a single cell. Science. 2002; 297(5584):1183–1186. [PubMed: 12183631]

36. Matlab watershed segmentation example. http://www.mathworks.com/products/demos/image/watershed/ipexwatershed.html

37. Carpenter AE, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. Genome Biol. 2006; 7:R100. [PubMed: 17076895]

38. Li G, et al. Segmentation of touching cell nuclei using gradient flow tracking. J Microsc. 2008; 231(1):47–58. [PubMed: 18638189]

39. Dean, J.; Ghemawat, S. Simplified data processing on large clusters. OSDI'04: Sixth Symp. on Operating System Design and Implementation; Google Research Publication, San Francisco, CA. 2004.

40. Wilkinson MHF, et al. Concurrent computation of attribute filters using shared memory parallel machines. IEEE Trans Pattern Anal Mach Intell. 2008; 30(10):1800–1813. [PubMed: 18703832]

41. Buades A, Coll B, Morel JM. A review of image denoising algorithms, with a new one. Multiscale Model Simul. 2005; 4:490–530.

42. Haworth CD, Petillot Y, Trucco E. Image processing techniques for metallic object detection with millimetre-wave images. Pattern Recogn Lett. 2006; 27(15):1843–1851.

Biographies

Quanli Wang received his MSc in computer science from Huazhong University of Science and Technology, Wuhan, China (1995) and his MSc in statistics from the University of Alberta, Alberta, Canada (1999). He worked as a software engineer at the Institute of Computing Technology, Chinese Academy of Science, Beijing, China between 1995 and 1997 and a mathematician at Halliburton Canada between 1999 and 2003. He was senior bioinformatician at the Institute for Genome Sciences & Policy, Duke University, between 2003 and 2009. Since then he has been with the Department of Statistical Science at Duke University. His research interests include high-performance computing in Bayesian statistics, bioinformatics, and image processing.

Mike West is the Arts and Sciences Professor of Statistical Science in the Department of Statistical Science at Duke University. He works on Bayesian statistics and scientific computation, including methodology development and applications of complex stochastic modelling in high-dimensional problems. He is an expert in time series analysis, forecasting, and computational statistics, and has published widely in areas including signal processing, finance, climatology, public health, biomedicine, and systems biology. He led the development of statistics at Duke from 1990 until 2001, has served in the establishment and on founding boards of several national research institutes including NISS and SAMSI in the USA, ISM in Japan, and IMS in UK, as well as President of the International Society for Bayesian Analysis (2009). He has advised over 50 primary PhD students and postdoctoral associates, most of whom are now in faculty, industrial, or governmental positions involving advanced statistical research.

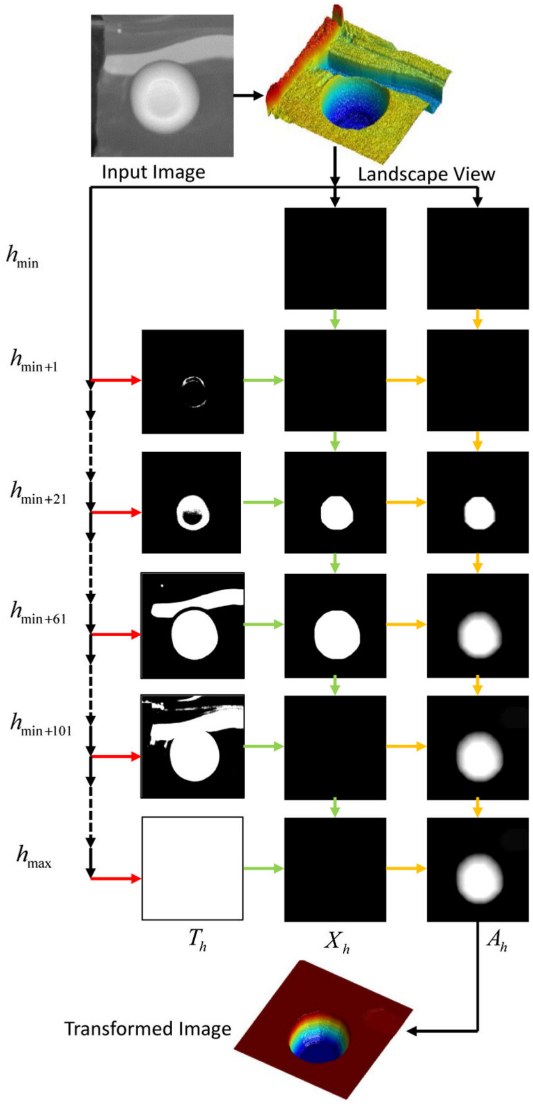


Fig. 1.

The flow chart of the model-controlled flooding framework.

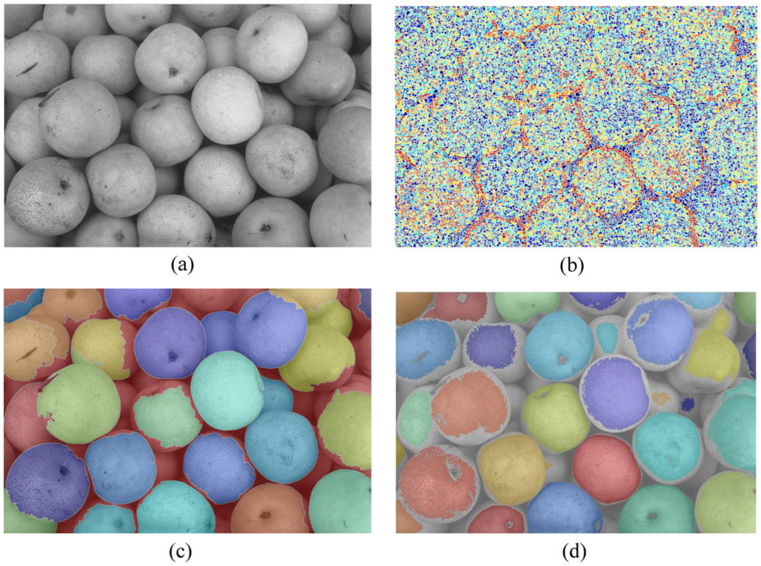


Fig. 2.

Pear data example, (a) original image, (b) watershed transform, (c) combination of marker-controlled watershed transform and morphological operations, (d) MCF size transform.

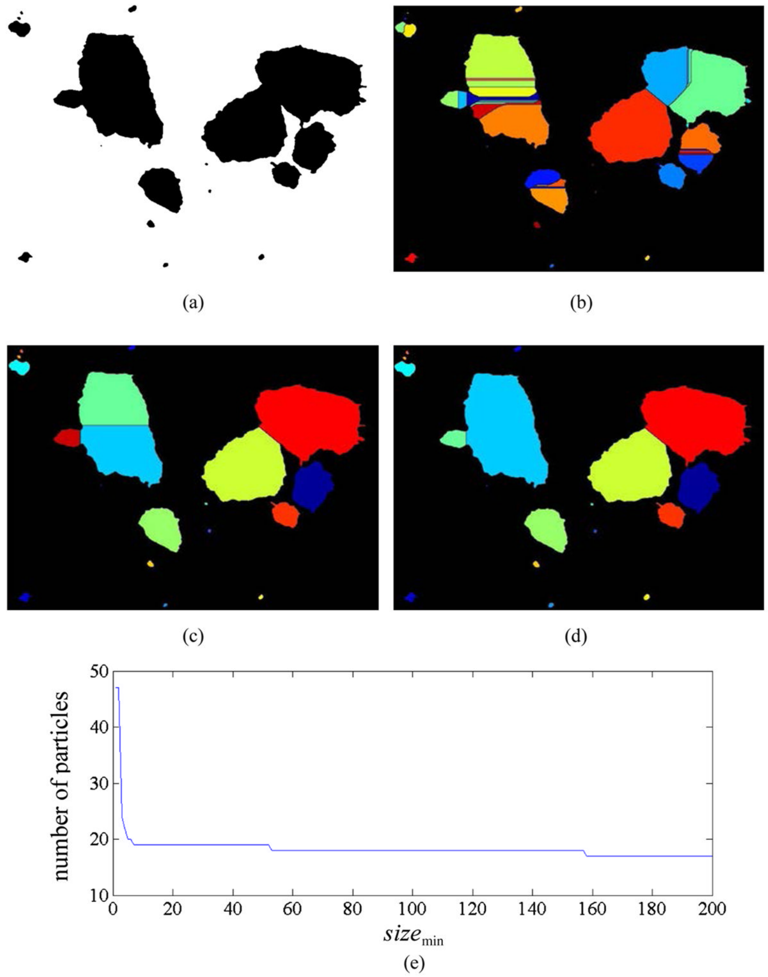


Fig. 3.

(a) input binary particle image, (b) watershed segmentation, (c) MCF size transform with minimum size 25, (d) MCF size transform with minimum size 100, (e) number of particles versus minimum size.

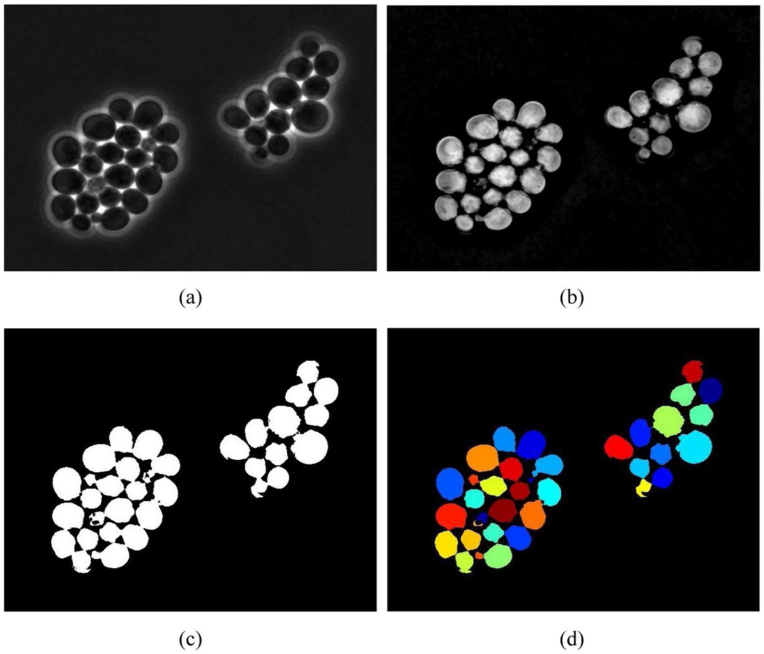


Fig. 4.

(a) Input budding yeast image; (b) contrast enhancement using MCF size transform; (c) extracted foreground; (d) final result using binary particle segmentation algorithm as in example 2.

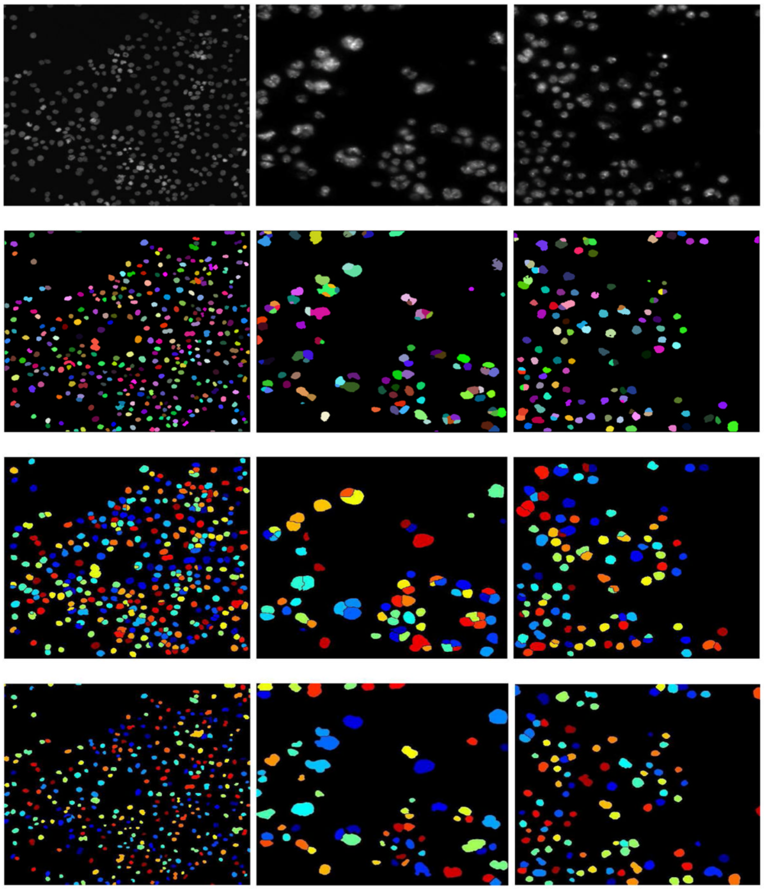


Fig. 5.

Segmentation results for selected human and fly data. Row 1: input images; row 2: Gradient Flow Tracking; row 3: CellProfiler; row 4: MCF size transform.

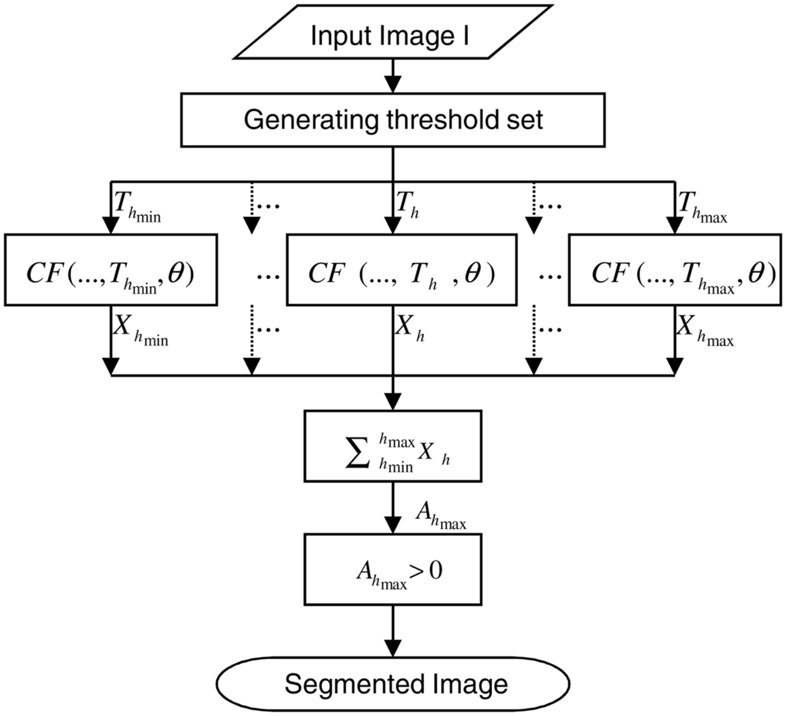
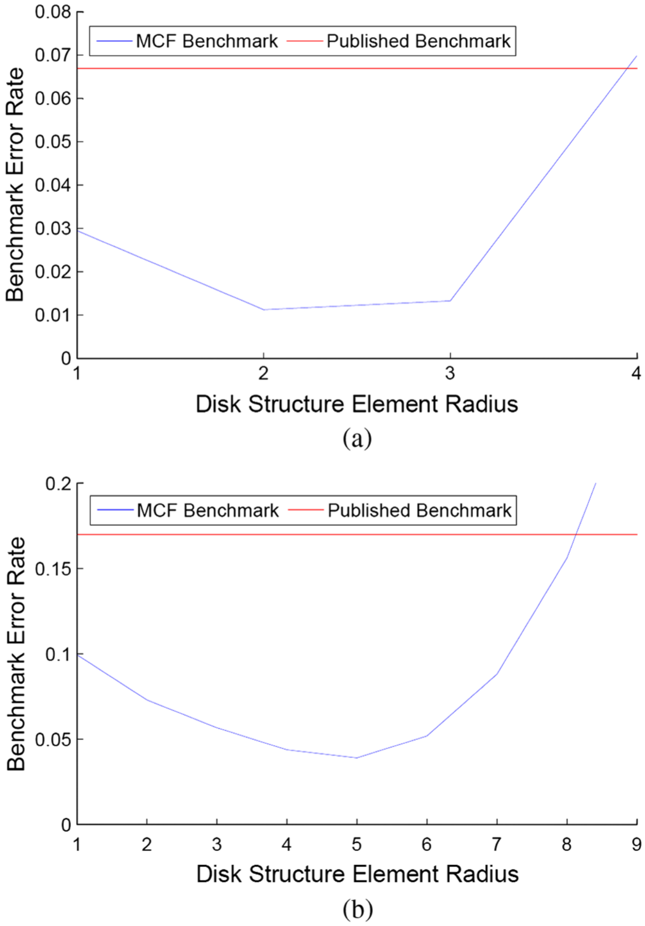


Fig. 6.

Flowchart for parallel MCF size transform implementation



(a)

(b)

Fig. 7.

Benchmarks versus tuning parameter “disk structure element radius” for (a) human and (b) fly (b) data sets. Blue lines are benchmarks from the new MCF algorithm and red lines are the published results from CellProfiler.

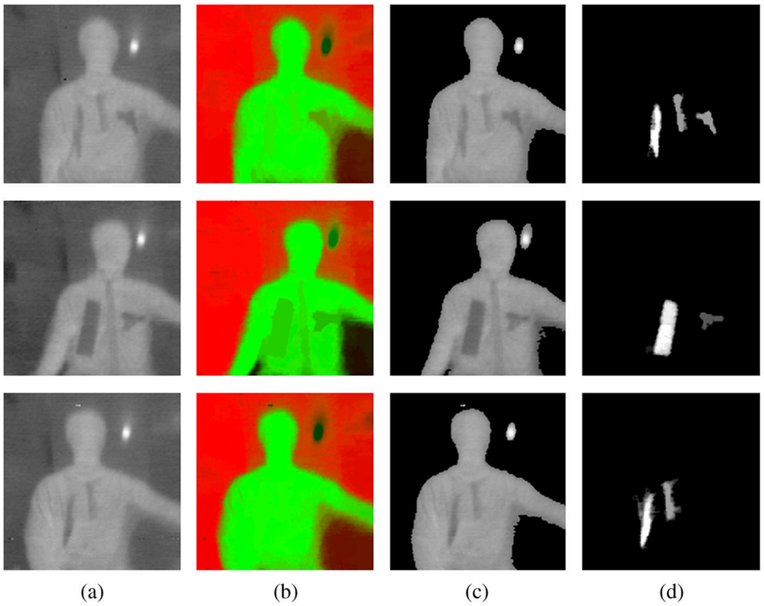


Fig. 8.

Detection and segmentation of concealed objects in terahertz images. Column (a): the original images; column (b): smoothed background and human body using size transform; column (c): identified background; column (d): identified concealed objects using size transform.

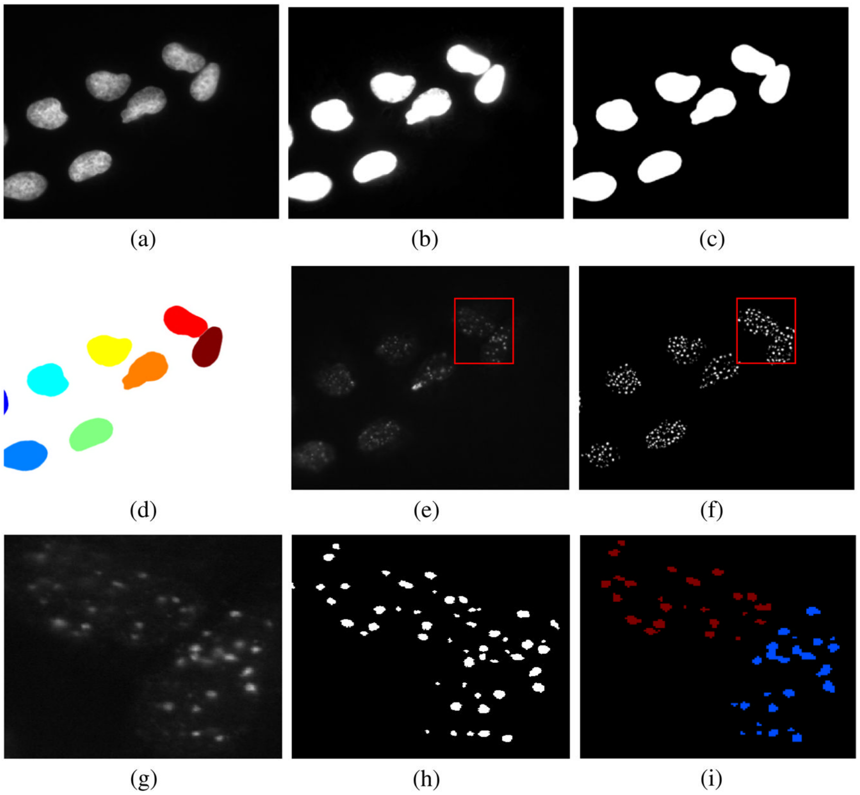


Fig. 9.

Speckle counting. (a) original nuclei image; (b) preprocessed image using size transform; (c) Otsu global thresholding of (b); (d) nuclei separation using size transform; (e) original foci image; (f) foci detection result using size transform; (g), (h) close-up images of (e) and (f) enclosed by the rectangles; (i) CellProfiler result.

Table 1

Benchmark analyses: Implementation details.

		Human HT29 Colon Cancer 1	**Drosophila KC 167 1(fly)**
Number of data sets		6	5
Images per data set		1	10
Total number of images		6	50
Method		**1** Apply the customized size transform to transform input images. **2** Use Rank Transform on hybrid images with default parameters threshold images. **3** Use the procedure described in example 2 for final binary segmentation.	
Parameters	Structure element radius	2	5
	Maximum cell size	5000	
	Minimum Highest Intensity value	0 (not used)	20
Benchmark calculations:		Compute absolute difference between the algorithm's count and the average of the humans' counts, then divide by the latter to obtain deviation from ground truth (in percent). The mean of these values over all six images is the final result.	Compute the algorithm's mean cell count over the set of 10 images. Calculate the absolute difference between this mean and the average of the humans' counts for the sample, then divide by the latter to obtain the deviation from ground truth (in percent). The mean of these values over all five data sets is the final result.

Table 2

Benchmark results.

Method	Human	Fly
CellProfiler	6.7%	17%
Gradient Flow Tracking	1.35%	82%
MCF	1.13%	3.9%