# Fast Marker Count Reduction for Watershed Transforms

**Bryan J. Mealy**

UCSC-CRL-02-38
December 12, 2002

Computer Engineering Department
University of California
Santa Cruz, CA 95064

## Abstract

Despite the Watershed Transform's (WST) tendency to oversegment, it has become a widely used method to segment images. In most published WST applications, gradient magnitude thresholding (GMT) is applied as a method to control WST oversegmentation. GMT is relatively fast and simple and reduces WST region count without losing important edge information in the original image. Region reduction via GMT, however, is confined to smooth areas of the image. In this paper we present a new method to control WST oversegmentation which we refer to as *marklet coupling filtering* (MCF). This novel technique controls WST oversegmentation in non-smooth image areas. The term *marklet* refers to a regional minima in the gradient image comprising of one pixel. Region count is reduced when two adjoining marklets are combined to form a single marker. Two different types of MCFs and are described. We demonstrate the ability of the MCF to control WST oversegmentation without compromising shape information in the original image.

The results show that the number of regions in the partitioned image are significantly reduced by applying MCF. These results are presented after a GMT application with a modest threshold value. The results also show that important edge information from the original image is preserved. Both qualitative and quantitative results are provided.

**Keywords:** watershed transform, oversegmentation, marker, regional minima.

# 1 Introduction

The Watershed Transform's (WST) known tendency to oversegment presents a major challenge when used for image segmentation. Oversegmentation reduces the average region size over the partition. This in turn reduces the probability that any one region represents a visually important feature in the image. In order to increase the probability that regions correspond to visually significant features, some form of region merging is applied after WST processing. Region merging algorithms, however, are computationally expensive and do not guarantee viable results.

Gradient magnitude thresholding (GMT) provides an automatic method of oversegmentation control for WSTs. The wide use of GMT in published WST applications attests to its efficacy in control oversegmentation. GMT, however, is limited to controlling oversegmentation in the smooth areas of the image. Techniques such as low-pass filtering of the original grayscale image [3] are commonly applied as an aid to reducing the generation of regions attributed to noise. These techniques, however, have the undesirable effect of reducing viable shape information from the original image.

This paper describes the technique of *marklet coupling filtering* (MCF) used to automatically reduce WST oversegmentation. Similar to GMT, MCFs are applied prior to the actual WST algorithm. MCFs are not dependent upon GMT and can be applied regardless of whether GMT is used. MCFs independently reduce the number of markers generated without significantly increasing the time or space complexity of WST preprocessing. The resulting watershed-generated partition is functionally equivalent to the partition generated by an unaltered WST application.

In this work, we use the algorithm of Beucher and Meyer [1] in conjunction with the preprocessing algorithm described by Dobrin *et al.* [2]. This particular WST algorithm generates a complete tessellation of the image and does not assign pixels to watershed lines. Implementation details are provided where appropriate but we assume the reader is familiar with the concepts of immersion simulation-based WSTs. Reference [4] provides a complete introduction to immersion simulation-based WSTs.

# 2 Immersion Simulation WST Overview

Fig. 1 shows the four steps required for an application of an immersion simulation-based WST. This diagram shows three steps of preprocessing followed by the actual WST. The first step generates a gradient image based on the original grayscale image. The next step designates certain portions of the image as regional minima which become the *prick-points* for the ensuing immersion simulation. The initial pixel queuing step adds pixels to the *ordered queue* to create a starting state for the WST. The final step is the actual WST.

## 2.1 Gradient Image Generation

The first step in WST processing generates a gradient image based on the original grayscale image. Each pixel in the grayscale image, $I$, is mapped to a value representing the discrete

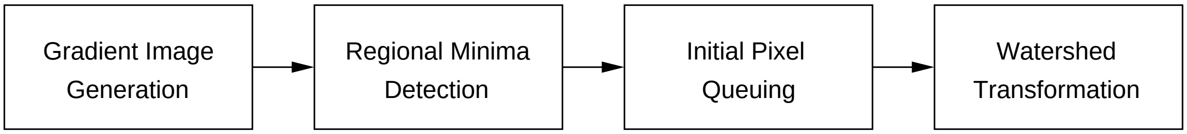


Figure 1: Process flow of the immersion simulation-based WST algorithm.

gradient magnitude at its location. This mapping forms the gradient image $X$, *i.e.*, $\nabla f : I \to X$. Each pixel $p$ in $I$ is mapped to $X$ by the function $\nabla f$ which approximates the gradient magnitude at $p$. Gradient magnitude thresholding is applied during the generation of the gradient image and essentially acts as a low magnitude clipping operation. Further description and discussion of gradient magnitude thresholding is found in [3].

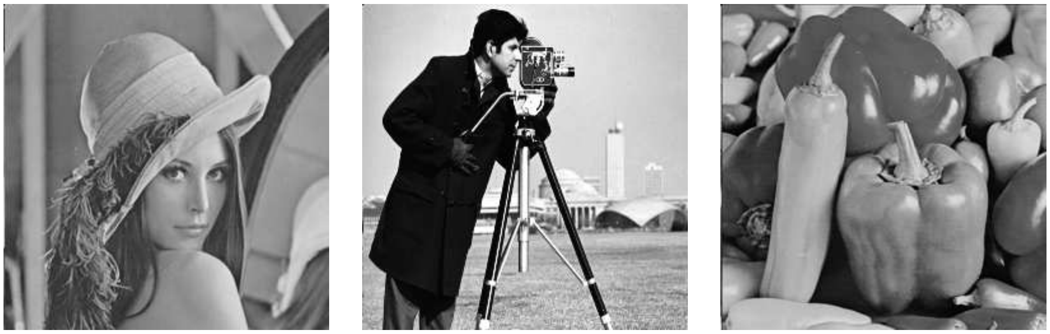
### 2.2 Regional Minima Detection

The application of the WST requires locating and labeling regional minima in the gradient image. Regional minima represent the prick-points used in immersion simulation-based WSTs and serve as seed points for the region growing process. Each prick-point spawns the growth of a catchment basin, and each catchment basin contains only one prick-point. Although prick-points can be chosen by an outside user, we limit the discussion to automatic regional minima detection techniques.

The first work by Dobrin *et al.* [2] presented an algorithm for regional minima detection. Regional minima are not explicitly assigned by the Dobrin's regional minima detection algorithm. In the Dobrin *et al.* approach, the algorithm instead locates and marks pixels that do not fit the definition of local minima. These pixels are labeled as *not a regional minimum*, or *NARM*s. The gradient image, $X$, is transformed into an image where each pixel is either a NARM or the initial value assigned by the algorithm. Pixels containing initial values are then considered regional minima. This procedure creates a "sea" of NARMs interspersed with "islands" of regional minima. This definition of regional minima renders the analogy of "prick-points" from immersion simulation models misleading since the minima are not constrained to being single pixels, or "points". Instead, a single regional minima can be as small as a single pixel or as large as a major portion of the image. In this paper, we use the term *markers* synonymously with regional minima.

## 3 Marklet Coupling Filters (MCF)

As shown in Fig. 1, the second step in applying a WST is the detection of regional minima, $M$. The resulting marker image, $I_M$, is a binary image where each pixel $p \in M_i$ or $p =$ NARM. Pixels with low gradient values correspond to smooth areas of the grayscale image and tend to cluster in large markers. This characteristic is magnified when gradient magnitude thresholding is applied. Most natural images contain large areas of pixels exhibiting low gradient values which are eventually grouped into large catchment basins. Consequently, large portions of the image become associated with a relatively small number of regional minima. Conversely, there tends



lena (256x256) tripod (256x256) peppers (256x256)

Figure 2: The test images.

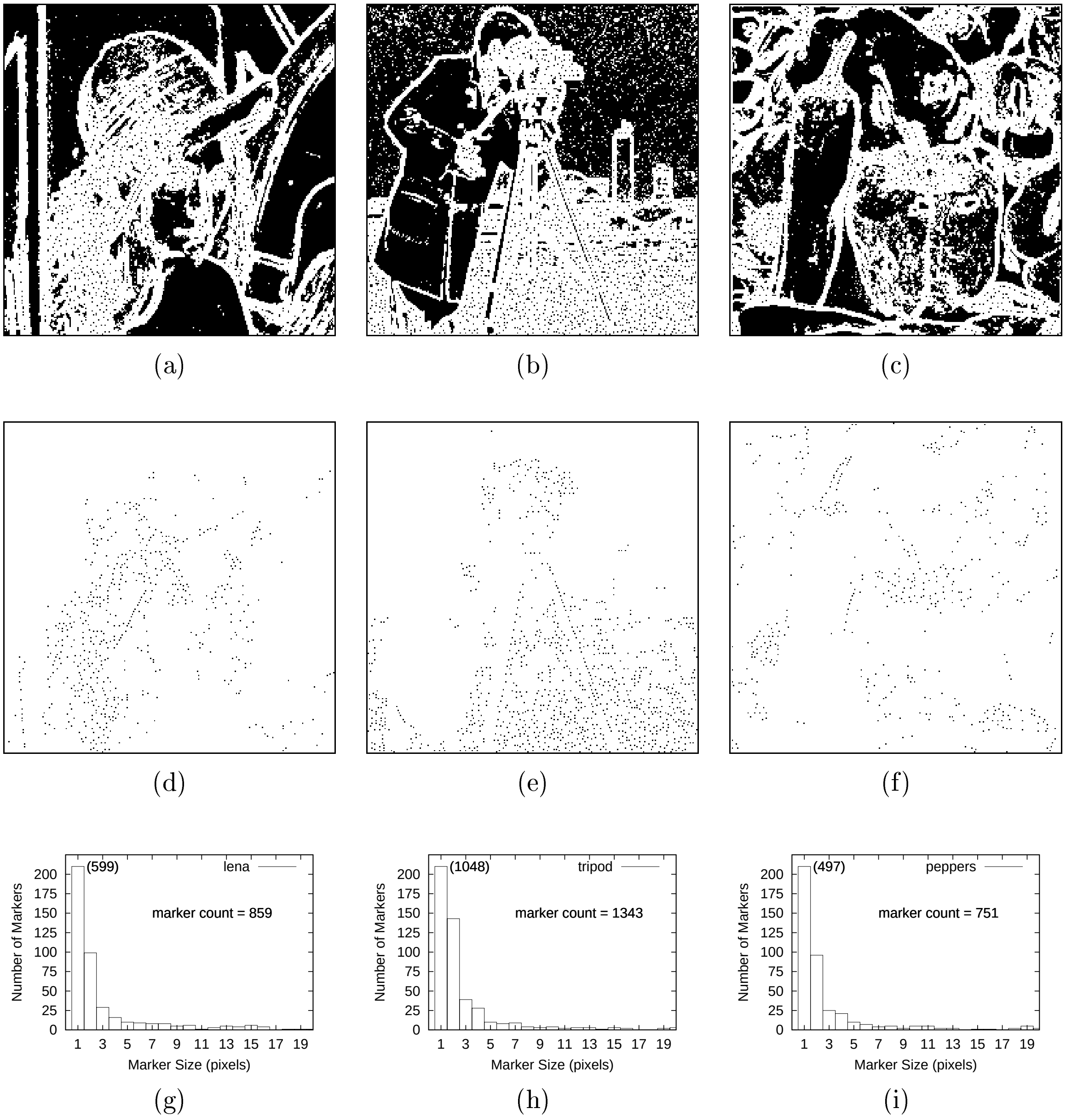
to be a high density of regional minima in areas of the image containing high spatial frequencies. The size of these markers tends to be smaller than markers found in smooth areas of the image. These smaller markers are generally less important than larger markers and can be combined with other small and local markers in the image without losing salient shape information in the final partition. This leads to a new approach for reducing the marker count prior to applying the actual WST algorithm.

The new technique targets only small markers which we refer to as *marklets*. A marklet, denoted by $\dot{M}$, is a regional minimum comprising of a single pixel. The marklet image, $I_{\dot{M}}$, is formed from markers in the marker image by $I_{\dot{M}} = \{M_i \mid \|M_i\| = 1 \text{ for } M_i \in I_M\}$ with $i = \{1, 2, \ldots, k\}$ where $k$ is the number of markers in the marker image. Thus, $I_{\dot{M}} \subseteq I_M$. Marklets that are located a short distance from other marklets are combined to form a single marker. Each of these combinations reduces the region count by one. Since no marker is removed from the image, edge information contained in the associated regions is retained. The number of pixels involved in the initial queuing process changes only slightly.

Fig. 2 shows the original lena, tripod, and peppers test images. Fig. 3(a)-(c) shows the marker images associated with the lena, tripod, and peppers test images, respectively. The marker image is a binary image where every pixel is either a marker or a NARM; the dark areas of these images represent the markers. Fig. 3(d)-(f) shows the marklet images associated with Fig. 3(a)-(c), respectively. The marklet image, $I_{\dot{M}}$, is a binary image where every pixel is either a marklet or a non-marklet[1]. Fig. 3(g)-(i) shows a partial histogram of marker cardinality for each image. The first entry in the histograms are not drawn to scale but actual counts are listed in parenthesis next to that entry. Each histogram shares the same general shape with marklets comprising a considerable portion of the total marker count.

We present two types of MCFs: D5 and D7, which are designed to work on the same image in two-step process. The D5 MCF provides an initial level of marklet reduction and can be followed by the D7 MCF, which provides a second level of marklet reduction. Fig. 4 shows that the D5 MCF must be applied prior to the D7 MCF. The D5 MCF is applied before the D7 MCF since the efficiency of both filters is based upon known spacing of the marklets in $I_{\dot{M}}$. Applying

[1] In this case, the non-marklet is either a NARM or a marker with $M > 1$.


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3: An example showing the markers associated with images lena, tripod, and peppers. (a)-(c) show markers as dark areas. (d)-(f) are the marklet images, $I_{\dot{M}}$, and (g)-(i) are partial histograms of marker sizes for the images of (a)-(c), respectively.

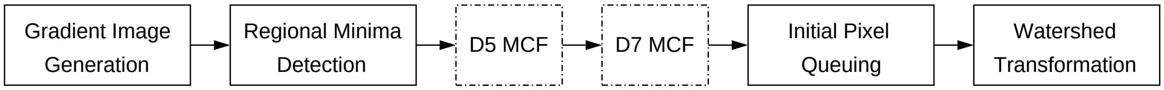


Figure 4: Modified process flow of the immersion simulation-based WST. This figures shows the stages where D5 and D7 MCFs are applied.

the D5 MCF does not require the subsequent application of the D7 MCF.

By definition, each marklet is bordered by eight NARMs which ensures that marklets are separated by a $D_8 \geq 2$. This also implies that any two marklets separated by a chessboard distance[2] of $D_8 = 2$ can be connected by placing a single marklet between them. After the application of the D5 MCF, remaining marklets are then separated by a $D_8 \geq 3$. This in-turn allows marklets separated by a distance $D_8 = 3$ to be connected by the addition of two pixels between them. The use of $D_8$ as a distance measure causes the region of support for the D5 and D7 MCFs to be 5x5 and 7x7, respectively. The "D" in the filter names represents the diameter of the filters as described below.

Fig. 5(a) shows the template used to apply a D5 MCF to a marklet located at location $X$. The template uses a 5x5 region of support centered about pixel location $p$. Eqn. 1 describes the operation of the D5 MCF. The D5 MCF is implemented by passing the template over each pixel in $I_{\dot{M}}$ in a method reminiscent of convolution. The implementation is efficient since not every template location with $D_8 = 2$ from pixel $p$ requires examination. Not examining every pixel location allows the templates of Fig. 5 to consider only pixels in the lower half of the templates; only shaded the pixels on the template edges are evaluated. The D5 MCF effectively adds a marklet to $I_{\dot{M}}$ between two existing marklets. The additional marklet transforms the two existing marklets into a marker, $M_{new}$, with $|M_{new}| \geq 3$.

The D7 MCF is similarly defined in Eqn. 2. The D7 filter is applied after the D5 MCF so that each marklet is necessarily separated by $D_8 = 3$. The coupling of marklets requires the addition of two marklets to join marklets on the perimeter of the 7x7 template centered at marklet $p$.

$$\text{if} \quad I_{\dot{M}}(p) \equiv \dot{M} \quad \text{then} \quad \begin{cases} I_{\dot{M}}(q) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{1,2\} \\ I_{\dot{M}}(r) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{3,4\} \\ I_{\dot{M}}(s) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{5\} \\ I_{\dot{M}}(t) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{6,7,8\} \end{cases} \tag{1}$$

$$\text{if } I_{\dot{M}}(p) \equiv \dot{M} \;\; \text{then} \begin{cases} I_{\dot{M}}(q), I_{\dot{M}}(a) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{1,2\} \\ I_{\dot{M}}(r), I_{\dot{M}}(b) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{3,4,5\} \\ I_{\dot{M}}(s), I_{\dot{M}}(c) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{6,7,8\} \\ I_{\dot{M}}(t), I_{\dot{M}}(d) \rightarrow \dot{M} & \text{when} \quad I_{\dot{M}}(j) = \dot{M} \quad \text{with} \quad j = \{9,10,11,12\} \end{cases} \tag{2}$$

[2] Chessboard and city-block distances are denoted by $D_8$ and $D_4$, respectively.

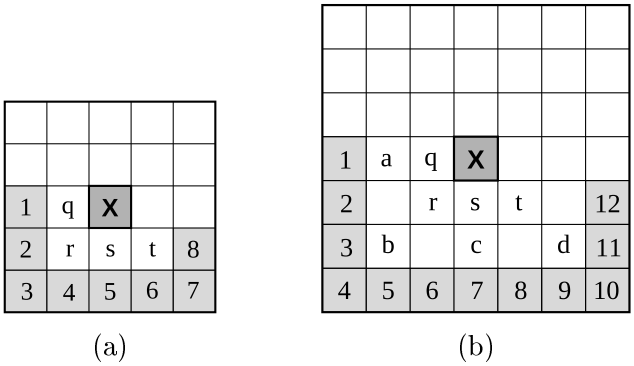


Figure 5: The 5x5 and 7x7 templates used by the D5 and D7 marklet coupling filters.

### 3.1 Example MCF Application

Fig. 6 demonstrates a D5 MCF application and shows it effects on the initial pixel queuing step. Fig. 6(a) shows a marklet image $I_{\dot{M}}$ with seven marklets, indicated by dark filled squares. Fig. 6(b) shows which pixels are involved in initial pixel queuing (*i.e.*, with hash lines). Fig. 6(c) shows the result of a D5 MCF with coupling pixels shown with gray fill. Fig. 6(d) shows the result of initial pixel queuing obtained from the MCF modified image. Three newly queued pixels appear as indicated by the presence of filled circles in the respective pixel locations. The two new markers created from the sets of marklets have cardinalities of three and five. The two marklets appearing in the upper-right corner of the image are unaffected by the D5 MCF application since the distance between them $D_8 > 2$. These two marklets are coupled if the D7 MCF is applied. Results for the D7 MCF are similar but are not shown.

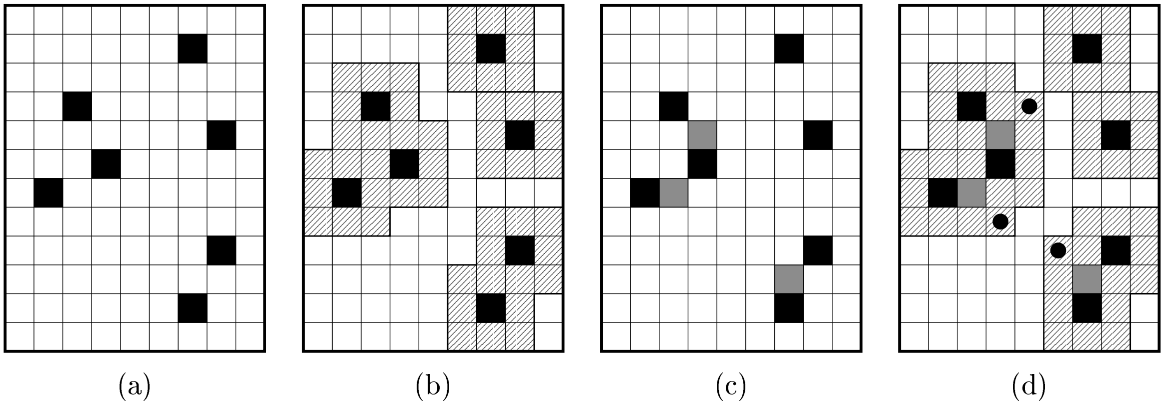


Figure 6: A demonstration of initial pixel queuing before and after MCF processing. (a) shows a marklet image containing seven marklets. The hashed pixels in (b) show which pixels in (a) are involved in initial pixel queuing. The lighter filled pixels in (c) show the marklets added by the application of the D5 MCF. The pixels with filled circles in (d) are pixels queued after the D5 MCF that were not queued in (b).

Applications of the D5 and D7 MCFs to the lena, tripod, and peppers test images are shown in Fig. 7 and Fig. 8, respectively. Images (a)-(c) in both figures show the affect that D5 and D7 MCFs have on the marklet images. It is evident that many of the marklets from the original marklet images of Fig. 3(a)-(c) were combined to form markers. Images in (d)-(f) of Fig. 7 and Fig. 8 show the borders of the resulting catchment basins after the WST application.

Fig. 7(g)-(i) and Fig. 8(g)-(i) show the histograms of marker sizes in pixel of the resulting markers. Markers one pixel in size are not shown to scale, but actual counts are listed in parenthesis besides the corresponding bar. Table 1 lists quantitative results.

The third and fourth columns of Table 1 list the total number of markers and marklets in the images, respectively. The fifth column lists the marklet totals as a percentage of the total marker count. Column six shows the drop in marklet count after D5 and D7 MCF application. Column seven lists the percent drop in region count resulting from the MCF applications relative to the total region count.

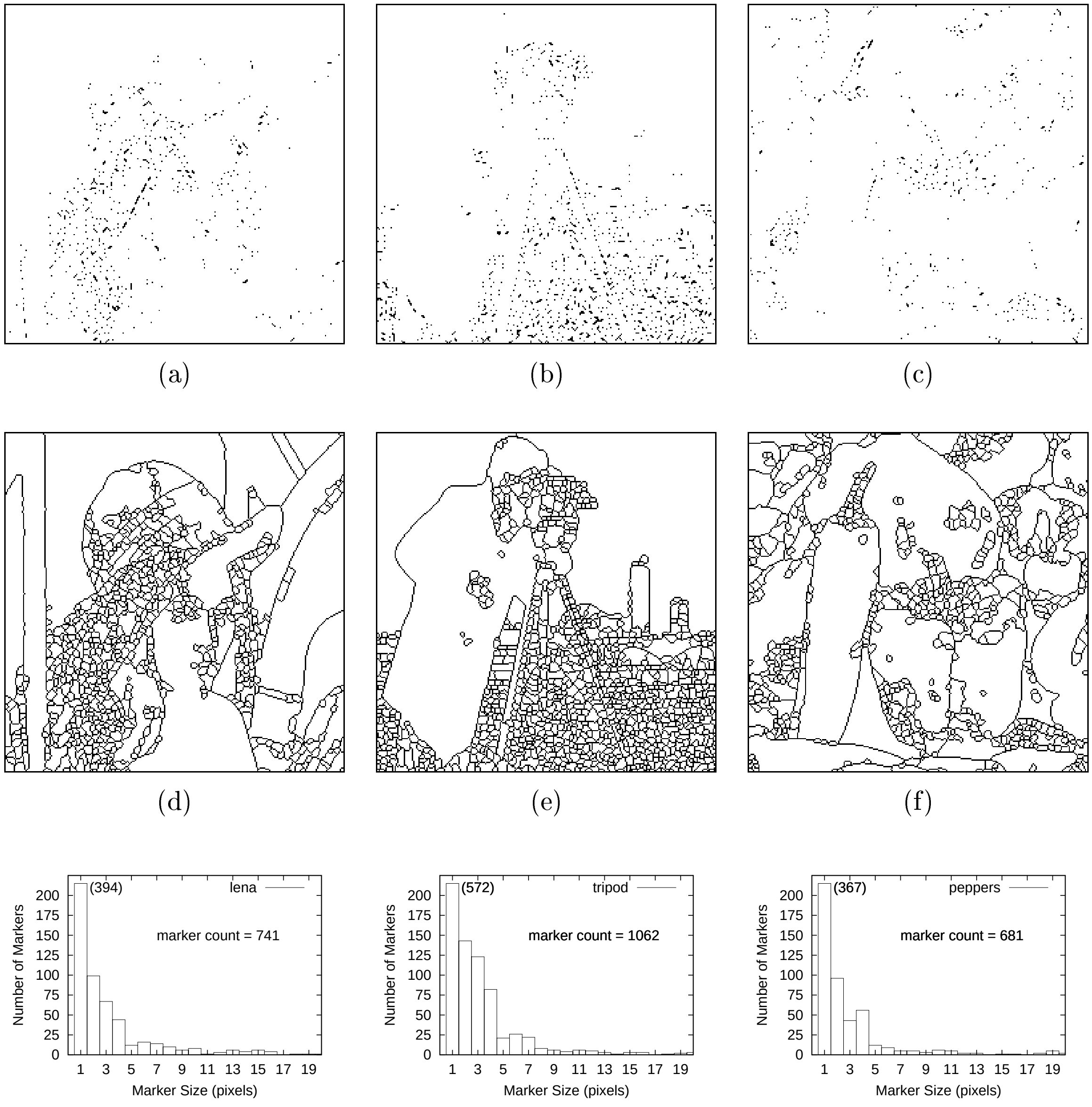
| image | Fig. 9 | # markers | # marklets | % marklet of total markers | # coupled marklets | % reduction in regions from baseline | comments |
|---|---|---|---|---|---|---|---|
| lena | (a) | 859 | 599 | 69.7 | - | - | baseline |
| | (d) | 741 | 394 | 53.2 | 205 | 13.7 | post D5 MCF |
| | (g) | 645 | 223 | 34.6 | 376 | 24.9 | post D7 MCF |
| tripod | (b) | 1343 | 1049 | 78.1 | - | - | baseline |
| | (e) | 1062 | 570 | 53.7 | 479 | 20.9 | post D5 MCF |
| | (h) | 897 | 295 | 33.1 | 754 | 33.2 | post D7 MCF |
| peppers | (c) | 751 | 498 | 66.3 | - | - | baseline |
| | (f) | 681 | 366 | 53.7 | 132 | 9.3 | post D5 MCF |
| | (i) | 617 | 245 | 39.7 | 253 | 17.8 | post D7 MCF |

Table 1: Results obtained from applying D5 and D7 MCF to test images.

Fig. 9 shows the region boundaries of the test images both before and after the application of the D5 and D7 MCFs. Fig. 9(a)-(c) are region borders with no MCF. Fig. 9(d)-(f) and Fig. 9(g)-(i) show region boundaries after D5 and D7 MCF, respectively. The images of Fig. 9(d)-(i) are the same region boundary images appearing in Fig. 7(d)-(f) and Fig. 8(d)-(f), repeated here for comparison. These images show that little shape information from the original image is lost while significant percentages of regions were removed from the busiest areas of the image.

### 3.2 MCF Space and Time Complexities

Applications of the D5 and D7 MCF require an extra scan through the image to search for marklets. Assume the image contains $m$ rows and $n$ columns with $N = m * n$. The image scan for the D5 MCF runs in $O([n-4]*[m-2])$ time which is the time required to search for marklets. When a marklet is located, searching for new marklets can advance by two column pixels since



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7: An example showing the changes in $I_{\dot{M}}$ after D5 MCF application. (a)-(c) show changes in the original marker image, (d)-(f) show the borders of the associated catchment basins after the D5 MCF application and (g)-(i) are partial histograms of marker sizes.

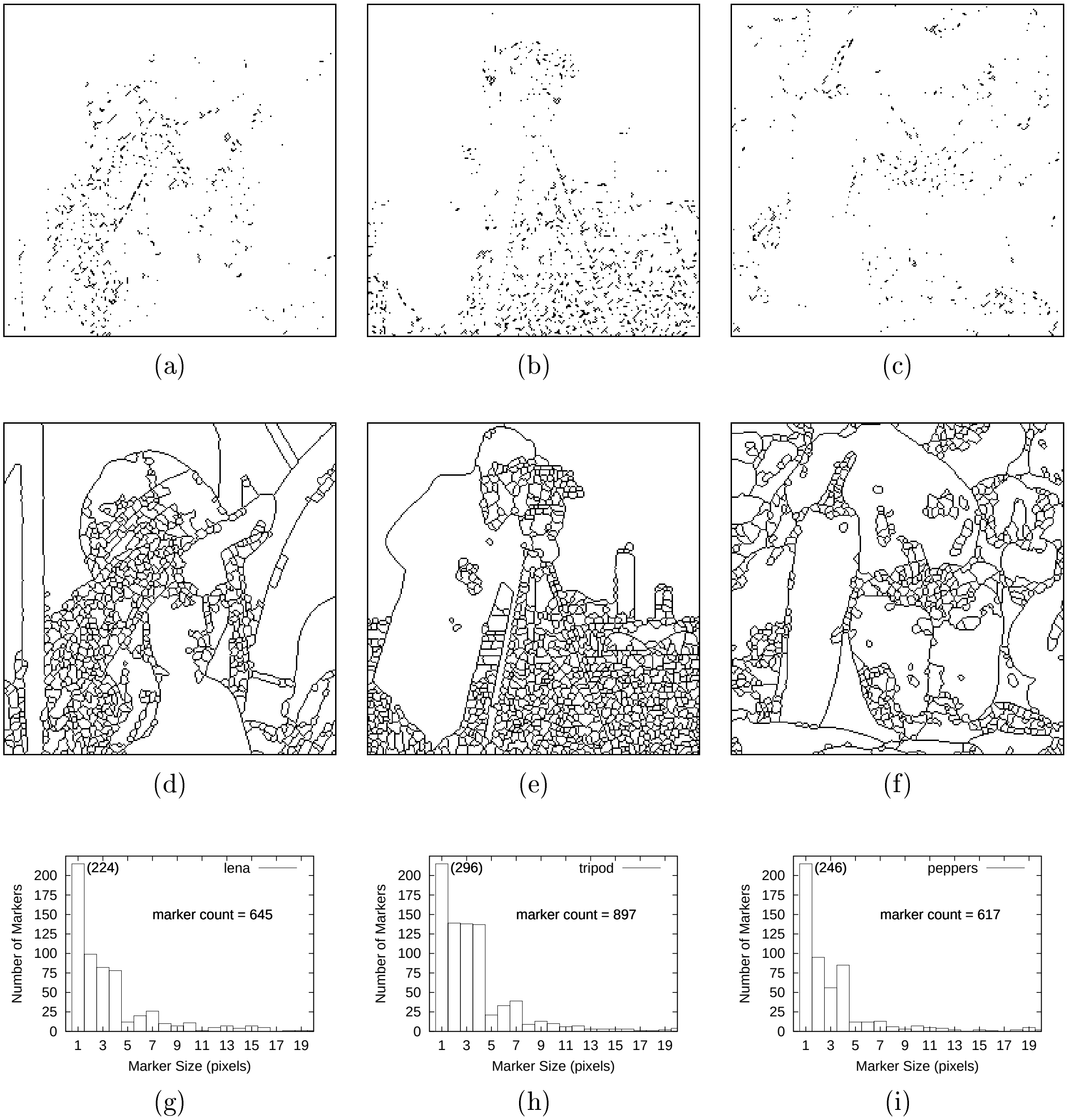
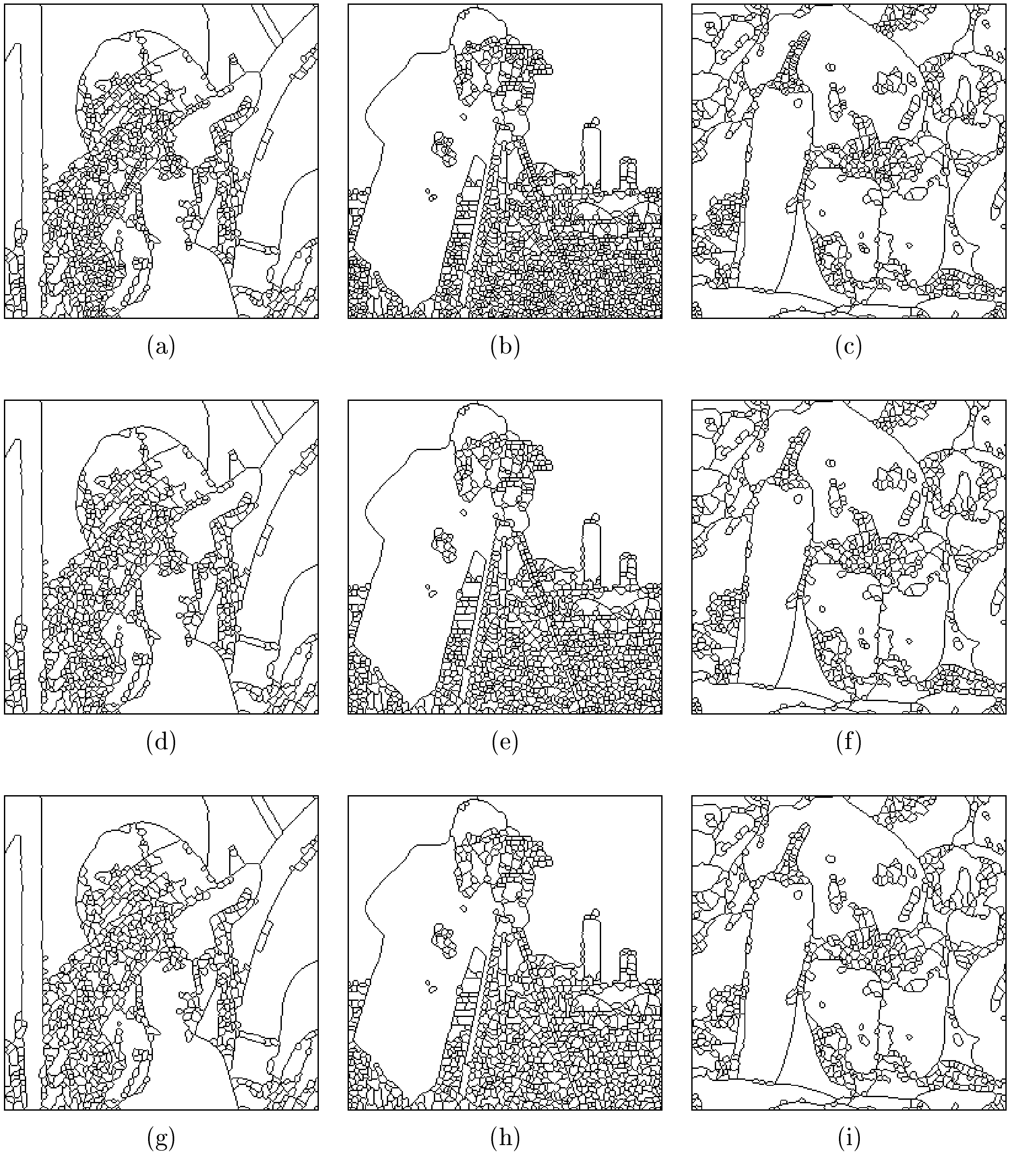


Figure 8: An example showing the changes in $I_{\dot{M}}$ after D7 MCF application. (a)-(c) show changes in the original marker image, (d)-(f) show the borders of the associated catchment basins after the D7 MCF application and (g)-(i) are partial histograms of marker sizes.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 9: An example of watershed region boundaries showing the effects of D5 and D7 MCF for the lena, tripod, and peppers images. (a)-(c) are the baseline images. (d)-(f) show the region boundaries after processing with the D5 MCF. (g)-(i) show the region boundaries after processing with the D7 MCF. Table 1 lists the corresponding quantitative results.

by definition no marklet is adjacent to another marklet. The worst case scan occurs when no marklets are present. For similar reasons, the D7 MCF runs in $O([n-6]*[m-3])$ time. Despite the extra processing time required for these MCFs, the overall $O(N)$ run-time complexity of the WST does not change. Application of both the D5 and D7 MCF require memory to store the marklet image. When both MCFs are applied, the same memory is used for both filters. In this case, the final memory requirement is increased by $N$.

## 4 Concluding Remarks

The results show that the number of regions in the final partition can be significantly reduced by the application of marklet coupling filters. This region reduction is focused in the spatially active regions of the image. Since no marker information is removed from the image, the MCFs are able to reduce region count without sacrificing visually important regions from the final partition. The MCFs are applied to the image after applying gradient magnitude thresholding with a modest threshold value. MCFs do not change the run-time complexity of the WST, and require only a slight increase in space complexity.

## References

[1] S. Beucher and F. Meyer. The Morphological Approach to Segmentation: the Watershed Transformation. In *Mathematical Morphology in Image Processing*, pages 443–481. Marcel Dekker, 1993.

[2] B. Dobrin, T. Viero, and M. Gabbouj. Fast Watershed Algorithms: Analysis and Extensions. In *Proc. Nonlinear Image Processing V*, volume 1769, pages 209–220. SPIE, February 1994.

[3] K. Haris, S. Efstratiadis, N. Maglaveras, and A. Katsaggelos. Hybrid Image Segmentation Using Watersheds and Fast Region Merging. *IEEE Transactions on Image Processing*, 7(12):1684–1699, December 1998.

[4] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.