Fast Region Merge Processing for Watershed Transforms

Bryan J. Mealy

UCSC-CRL-02-39
December 12, 2002

Computer Engineering Department
University of California
Santa Cruz, CA 95064

Abstract

Controlling the Watershed Transform's (WST) tendency to oversegment images is a major concern in published WST applications. Gradient magnitude thresholding (GMT) is the primary tool used to control WST segmentation due to its relatively fast and simple implementation. In this paper, we describe a new method to control WST oversegmentation which we refer to as *drainage simulation-based region merging*. The drainage simulation is applied after the WST and provides a mechanism for the fast merging of regions without requiring the updates of existing region descriptors. The new technique controls WST oversegmentation in the image areas containing the highest spatial frequencies. Similar to GMT, applying drainage simulations are automatic in that they require no outside user intervention.

The results show that the region count in the partitioned image is significantly reduced by utilizing drainage simulation-based region merging. These results are produced after an application of GMT using a relatively low threshold value. The results likewise indicate that important edge information from the original image is not compromised by the drainage simulation-based region merging.

Keywords: watershed transform, oversegmentation, gradient magnitude thresholding, region descriptors.

1 Introduction

Most published Watershed Transform (WST) applications apply some form of region count reduction (RCR) in an effort to remove trivial regions from the partition. The WST's tendency to oversegment has consequently produced many approaches to the RCR problem. Though these published techniques vary significantly in their approach, they share the goal of retaining only *salient*[1] regions in the partition.

The approach taken by most published WST applications to control oversegmentation is to apply some form of automatic region count reduction prior to formal region merge processing. Gradient magnitude thresholding (GMT) is the primary technique used for this purpose because of simple implementation and known efficacy. Reducing region count prior to applying formal region merging is desirable because region merge algorithms are computationally expensive and do not guarantee viable results. Drainage simulations provide another automatic region count reduction technique that is applied prior to formal region merging processing.

Complexity in the formal region merging processing occurs in three areas: 1) region descriptor updating, 2) the measure of similarity between adjacent regions calculation, and 3) the monitoring of region-merge termination criteria. Each of these areas requires attention after merging two regions during the formal region merge processing. In drainage simulation-based region merging, attention need not given to any of these areas. Required region descriptors can be calculated after drainage-simulation based merging is completed.

Published region count reduction algorithms are primarily concerned with creating meaningful partitions from segmented images without being overly complex. The general approach is to locate and merge a region pair that is more similar than any other region pair in the image. The similarity measure applied is based on any number of region characteristics. Probably the most widely used region characteristic is the average pixel intensity value of region pixels [8, 9, 12, 13, 14]. Another commonly used characteristic considers variations of heights associated with regions and region boundaries [2, 3, 6, 10]. Some of the more complicated techniques include *relaxation labeling* to draw upon local pixel information and constraint propagation to control region merging [7]. Obviously, as techniques become more complex, their running times are also increased.

Another common approach to decreasing the run-time required to for RCR processing is not considering all region pairs as merge candidates [9, 11]. This approach uses specialized data structures to support the topology of regions that have higher probabilities of being merged. The drawback of this type of approach is that it requires the creation and maintenance of extra data structures in addition to the main data structure to facilitate the region merge algorithm. The merging of two regions subsequently requires updates to both data structures which compromises the run-time savings obtained by not considering all regions in the merge processing.

2 Problem Formulation

At this point, the WST was applied to generate the initial watershed image, WS_X. This

[1] In this context, a salient region differs significantly from its neighboring regions based on some characteristic.

image is a complete tessellation derived from the gradient image, X, which is in-turn based upon the grayscale image, I. The regions in WS_X form a partition of I. The term $\Delta_K(I)$ denotes a partitioned image so that $\Delta_{K_o}(I) \equiv WS_X$ with K_o denoting the number of regions in $\Delta_{K_o}(I)$. Similarly, $\Delta_{K_\phi}(I)$ and $\Delta_{K_\omega}(I)$ are used to indicate the resulting partitions after the drainage simulation and region merging, respectively. Each partition is represented by a set of regions, *i.e.*, $\Delta_K(I) = \{R_1, R_2, \ldots, R_K\}$ and $\Delta_{K_o}(I) = \bigcup_{i=1}^{K} R_K$. Regions in the partition represent closed contours and there are no overlapping regions, *i.e.*, $R_m \cap R_n = \emptyset$, $\forall m, n \in \{1, 2, \ldots, K\}$ and $m \neq n$. Each region in the partition contains an associated label, k, shared by every pixel p_i the region, *i.e.*, $\forall R_k \in \Delta_K$, if $WS_X(p_i) \in R_k$ then $WS_X(p_i) = k$, $\forall WS_X(p_i) \in R_k$ with $i = \{1, 2, \ldots, |R_k|\}$. The number of regions in $\Delta_{K_\phi}(I)$ and $\Delta_{K_\omega}(I)$ are K_ϕ and K_ω, respectively.

2.1 Processing via Regional Adjacency Graph

Partition representation and manipulation is facilitated by a regional adjacency graph (RAG) [4]. The RAG associated with the partition is defined by $\mathcal{G} = (V, E)$. Each of the K regions in the image is represented by a graph node with $V = \{1, 2, \ldots, K\}$. Adjacency between two regions $R_m, R_n \in V$ is indicated by the existence of an edge $E(R_m, R_n) \in E$ between those two regions. Edges are assigned a single value indicating the similarity between the two regions they connect. This value is referred to as a *similarity measure* and is assigned by a similarity function. The general RAG processing paradigm is to locate the edge with the highest similarity measure and merge the two regions it connects.

The two main components of RAG construction are the assignment of pixels to nodes and the creation of edges between adjacent regions. Each node in the graph has an *identity* [1] which contains both topological and region characteristic information, or *descriptors*. Region topology information indicates adjacency to other regions while region characteristics describe relevant region qualities. Each pixel in WS_X contains a label, k, indicating the region to which it belongs. Processing at each pixel establishes both region membership and adjacency to other regions. When RAG construction is complete, each node contains a list of pixels in the associated region and a list of adjacent regions.

2.2 The Region Merge Paradigm

The number of regions in the partition is reduced by combining two regions to form a single region. This region combining process is referred to as region merging or region-merge processing. The new region created from merging two regions contains all the pixels in the pre-merged regions.

Merging two regions requires a considerable amount of processing to update RAG to reflect changes generated by the merging. Merging two regions requires identity updating for the two regions, as well as every region adjacent to those two regions. The $O(|V|)$ running time of these updates is based on a worst case merging scenario where V is number of nodes[2] in the RAG.

[2]Each node represents a region in the partition.

To facilitate the discussion of region merging, we define a region-merge operator MRG(). The arguments of MRG() are the two regions to be merged. Applying this operator results in the creation of a new region consisting of each pixel and the corresponding adjacency information existing in the individual regions prior to merging. The MRG() operator is applied by the RAG to incorporate changes resulting from combining two regions. Fig. 1 lists the main functions of the MRG() operator.

- Determine region to be subsumed
- Update neighbors of subsumed region to be adjacent to consuming region
- Update consuming region to be adjacent to neighbors of subsumed region
- Append pixel list of subsumed region to consuming region
- Update size for combined region
- Update region descriptors for combined region

Figure 1: Operations associated with the MRG() operator.

3 Region Reduction via Drainage Simulations

The introduction of immersion simulations made WST implementations practical. Flooding algorithms not only significantly reduced the running time of WSTs, but also improved the accuracy of their results. These algorithms are based on simple and eloquent concepts which is the primary reason for the significant improvement in results.

The watershed lines in the partition generated by the WST contains edge information from the original image. The WST's tendency to oversegment is somewhat reduced by applying a gradient magnitude thresholding to the gradient image, X. Gradient magnitude thresholding reduces WST oversegmentation without changing the run-time or space complexities of the WST algorithm. These qualities form the basis for the technique of region reduction via drainage simulations.

The concept of immersion simulations is developed by considering an image to be a topographic relief. Pixel values are considered elevations to give an image 3-dimensional qualities. This relief is then "pierced" in designated areas to act as seed points to the catchment basins. The relief is immersed into water and is slowly flooded. Catchment basins form as immersion progresses and water flows in from the piercing points. The process is complete when immersion depth exceeds the highest elevation in the image.

The concept of drainage simulations can be envisioned in two different ways. The first method is modeled as a process of slowly removing the previously flooded relief from water. In this model, *drainage basins*[3], which are the analog of immersion simulation catchment basins, form as land masses emerge as water levels recede. The second method to envision the concept

[3]Some authors use *drainage basin* synonymously with catchment basin. In this paper, catchment basins are associated with WSTs and drainage basins are associated with drainage simulations.

of drainage simulations is in the terms of the immersion simulation. In this method, relief is flipped over before starting the immersion process. Both of these methods are described below.

The first method to envision drainage simulations begins when the relief is completely flooded. With the relief immersed, we add *drain-points* to the highest elevations of the image. Similar to prick-points, holes are inserted into the relief at designated areas in the image. Whereas prick-points occur at regional minima, drain-points occur at regional maxima. The relief is then slowly removed from the water. The drain points allow water levels in the higher elevations of the image to grow as air fills into those portions of the relief. Each drain-point seeds the development of a drainage basin. Consequently, large drainage basins associate with higher image elevations and small drainage basins tend to cluster in lower elevations.

The second method to envision drainage simulations involves an inversion of the original relief. In this method, the highest points on the non-inverted relief become the lowest points on the inverted relief. These new low-points become the prick-points of the immersion simulation. When the relief is immersed into water, the lower elevations, which were the higher elevations in the non-inverted image, form the larger catchment basins. Immersion of the inverted relief continues until all land masses are submerged.

The similarity of these two descriptions to the standard immersion simulation allows the drainage simulation to be implemented via the immersion simulation-based WST algorithm. The only change required is to reverse the pixel values associated with the gradient image. Multiple WST applications to a single image is not uncommon in practice [5]. The drainage simulation presented in this dissertation requires only one extra WST application.

The partition generated by the the drainage simulation, iWS_X, is used to direct initial region merge processing. Two adjacent regions in WS_X that are members of the same region in iWS_X are merged. By the nature of the drainage simulation, merging occurs in only the active areas of the image. This portion of the region merge processing is fast because searching for the best merge candidate is not required. Moreover, update of region descriptors and recalculation of similarity measures are not required.

3.1 Drainage Simulation-based Region Merging

Region merging based on drainage simulations is divided into three steps as shown in Fig. 2. The initial step generates an inverse gradient image based upon the original gradient image. The drainage simulation is applied to the inverse gradient image in the second step. The final step merges regions in WS_X that share the same region in iWS_X. The following sections describe these steps.

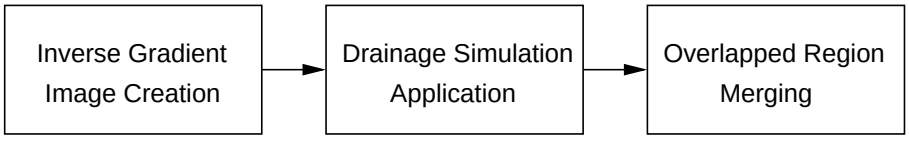


Figure 2: Application of the drainage simulation requires three steps.

3.2 Inverse Gradient Image Creation

The inverse gradient image, iX, is created from the gradient image X. This gradient image is not modified by gradient magnitude thresholding. Eqn. 1 is applied to all pixels to transform X to iX. Similar to T_∇ of the GMT equation, iT_∇ is a threshold value that controls the size of the various regions generated. In this case, larger regions form as iT_∇ is decreased.

$$iX(p) = \begin{cases} 0, & \text{if } X(p) < iT_\nabla \\ |X(p) - iT_\nabla|, & \text{otherwise} \end{cases} \tag{1}$$

Fig. 3(d)-(f) are the inverse gradient images associated with the lena, tripod, and peppers test images, respectively, which were generated with $iT_\nabla = 32$. These images are effectively negatives of the gradient images. When the relief is inverted, the dark areas of Fig. 3(d)-(f) are the low-points of the relief according to the second method used to envision the drainage simulation process.

3.3 Drainage Simulation Implementation

Transforming the gradient image into the inverse gradient image allows the drainage simulation to be implemented as an immersion simulation. Thus, the WST is applied to iX to generate iWS_X.

Fig. 3 show examples of drainage simulations for three test images. Fig. 3(a)-(c) show the original lena, tripod, and peppers test images. Fig. 3(d)-(f) show the inverse gradient images for test image of Fig. 3(a)-(c), respectively. Once X is transformed to iX, a standard WST is applied to iX. Fig. 3(g)-(i) show the inverse watershed images, iWS_X, associated with each test image. These images clearly show that image areas with higher spatial frequencies are grouped into larger regions. These areas become larger as iT_∇ decreases, and smaller as iT_∇ increases in a manner similar to, but opposite of, T_∇ used in gradient magnitude thresholding. Region counts for the image of Fig. 3(g)-(i) are listed in Table 1.

3.4 Overlapped Region Merging

Two watershed images, WS_X and iWS_X, were created by an immersion and drainage simulation, respectively. The watershed image, WS_X, retains the shape information from the original image. The inverse watershed image, iWS_X, contains regions used to control this portion of the region merge process.

The initial watershed image, WS_X, is referred to as $\Delta_{K_o}(I)$ and contains a set of K_o regions so that $\Delta_{K_o}(I) = \{R_1, R_2, \ldots, R_{K_o}\}$. The inverse watershed image, iWS_X, is referred to as $\Delta^\star_K(I)$ and contains a set of K_ϵ regions with $\Delta^\star_{K_\epsilon}(I) = \{R_1, R_2, \ldots, R_{K^\star_\epsilon}\}$. Overlapped regions, denoted by R_π, are defined as follows:

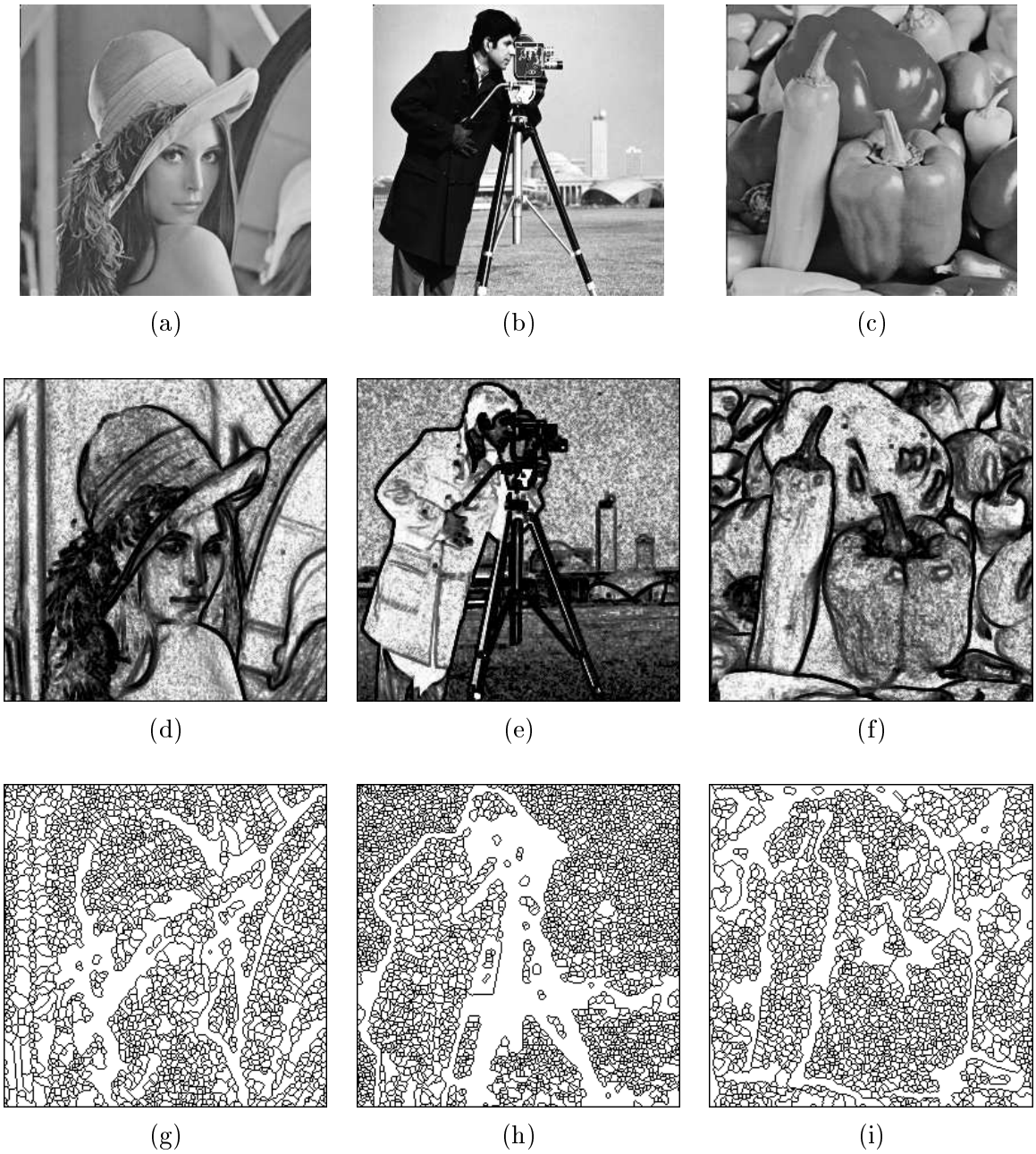


Figure 3: An example showing inverse gradient images and their associated drainage simulation partitions. (a)-(c) show the lena, tripod, and peppers test images of dimensions 256x256. (d)-(f) show the inverse gradient images of lena, tripod, and peppers, respectively. For these images, the darker areas represent the *lower* elevations of the relief. The boundaries of the associated region images that are generated by the drainage simulation are shown in (g)-(i).

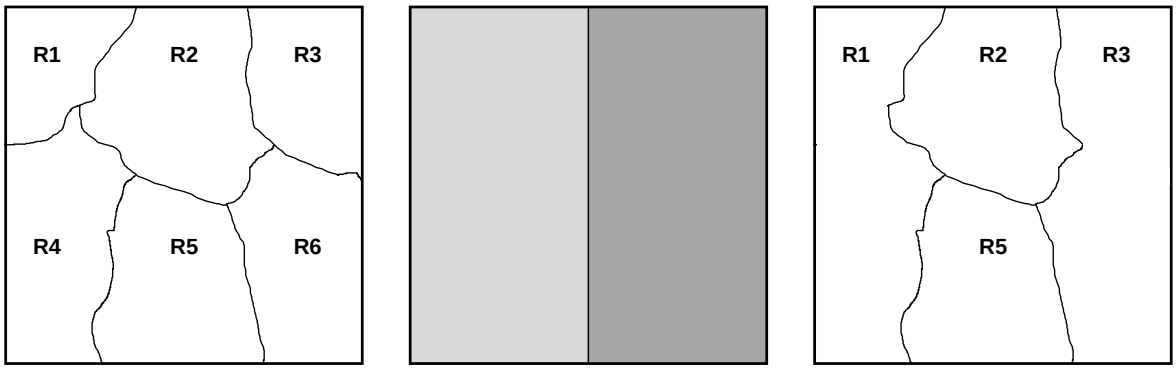


Figure 4: An example demonstrating overlap region merging. (a) shows a partition with six regions. (b) shows an inverse watershed image with two regions. The drainage simulation combines region pairs (R1-R4) and (R3-R6) since these pairs share the same region in (b). Regions R2 and R5 would not be combined since they overlap two regions in (b). (c) shows the partition after regions R4 and R6 are consumed by regions R1 and R3, respectively.

> **Definition 1**: An *overlapped* region, R_π, is any region $R_m \in \Delta_{K_o}(I)$ with every pixel, p, contained in R_m is a member of a single region in $\Delta^\star_K(I)$, *i.e.*, $R_\pi \equiv R_m \in WS_X$ such that $\forall p_i \in R_m$ with $i = \{1, 2, \ldots, |R_k|\}$, $iWS_X(p_1) = iWS_X(p_2) = \ldots = iWS_X(p_{|R_k|})$. The region label from iWS_X associated with some region $R_m = R_\pi \in WS_X$ is denoted by $R^m_\pi(k)$.

Regions are designated as overlapped during RAG creation. As pixels in WS_X are examined to establish their region membership, the corresponding pixels in iWS_X are also examined. Each RAG node representing an overlapped region also designates the region label in iWS_X of the region it overlaps.

Each region in the RAG is examined to see if it is overlapped. Regions adjacent to each overlapped region in the RAG are examined to see if they, too, are overlapped. Any two adjacent regions that are overlapped and share the same overlapped region label, $R_\pi(k)$, from iWS_X are merged with an application of MRG(), *i.e.*, if R_m, $R_n \in WS_X$ with $R_m \equiv R_n \equiv R_\pi$ and $R^m_\pi(k) = R^n_\pi(k)$, then MRG(R_m, R_n).

Fig. 4 demonstrates the region combining of the drainage simulation. Fig. 4(a) is a region image[4] containing six regions. Fig. 4(b) is the inverse watershed image generated from the same source as the region image of Fig. 4(a). The drainage simulation combines region pairs (R1-R4) and (R3-R6) because the pairs share the same regions in the inverse watershed image of Fig. 4(b). Regions R2 and R5 of Fig. 4(a) are not combined since they overlap two different regions in Fig. 4(b). Fig. 4(c) shows a post-drainage-simulation region count of four after regions R4 and R6 were consumed by region R1 and R3, respectively.

The results of drainage simulation-based region merging is shown in Fig. 5. The top row

[4]The region image is referred to as an *object membership map* by some authors.

of images in Fig. 5 show the boundaries of regions in $\Delta_{K_o}(I)$ for the test images. The second, third, and fourth rows show the region boundaries for Δ_{K_ϕ} with T_∇ values of 32, 25, and 18, respectively. Table 1 provides quantitative data for the image of Fig. 5. The third column of Table 1 shows the T_∇ values for the image with 255 indicating the baseline pre-drainage simulation image Δ_{K_o}. The fourth column lists the region counts for iWS_X. The fifth and sixth columns of Table 1 list the number of overlapped regions, R_π, and the number of these regions that were merged, respectively. The seventh and eighth columns list the region count of $\Delta_{K_\phi}(I)$ and the percent reduction in region count from the baseline image.

Fig. 6 shows how Δ_{K_ϕ} is affected by changes in T_∇. Fig. 6(a) shows the number of overlapped regions that are merged as a function of T_∇. Fig. 6(b) shows the number of overlapped regions that were merged as a percentage of the initial number of regions in the image.

image	Fig. 3	T_∇	N_{regs} iWS_X	# R_π	N_{mrgd} R_π	N_{regs} Δ_{K_ϕ}	% reduction in regions from baseline	comment
lena	(a)	255	-	-	-	645	-	baseline
	(d)	32	1935	154	94	551	14.7	-
	(g)	25	1800	254	185	460	28.7	-
	(j)	18	1670	352	289	356	44.8	-
tripod	(b)	255	-	-	-	817	-	baseline
	(e)	32	2715	298	225	672	17.8	-
	(h)	25	2472	444	367	530	35.1	-
	(k)	18	2262	586	497	400	51.0	-
peppers	(c)	255	-	-	-	617	-	baseline
	(f)	32	1933	146	70	547	11.4	-
	(i)	25	1823	218	136	481	22.0	-
	(l)	18	1738	295	205	412	33.2	-

Table 1: Results from drainage simulated merging of Fig. 3.

3.5 Drainage Simulation-based Region Merge Complexity

The analysis of drainage simulation-based region merging is divided into two parts: 1) iWS_X creation and, 2) region merging. Inverted watershed image creation requires extra storage for both the inverted gradient image and the inverse watershed image. Since the simulation is based on a standard WST, previously allocated image objects required by the WST are reused during the drainage simulation. Consequently, space complexity is increased by N, the number of pixels in the image. Region merging requires no extra memory since merging is already required. Space complexity is increased only slightly by the drainage simulations.

Creation of iX requires one scan through the image for a time complexity of $O(N)$. Though

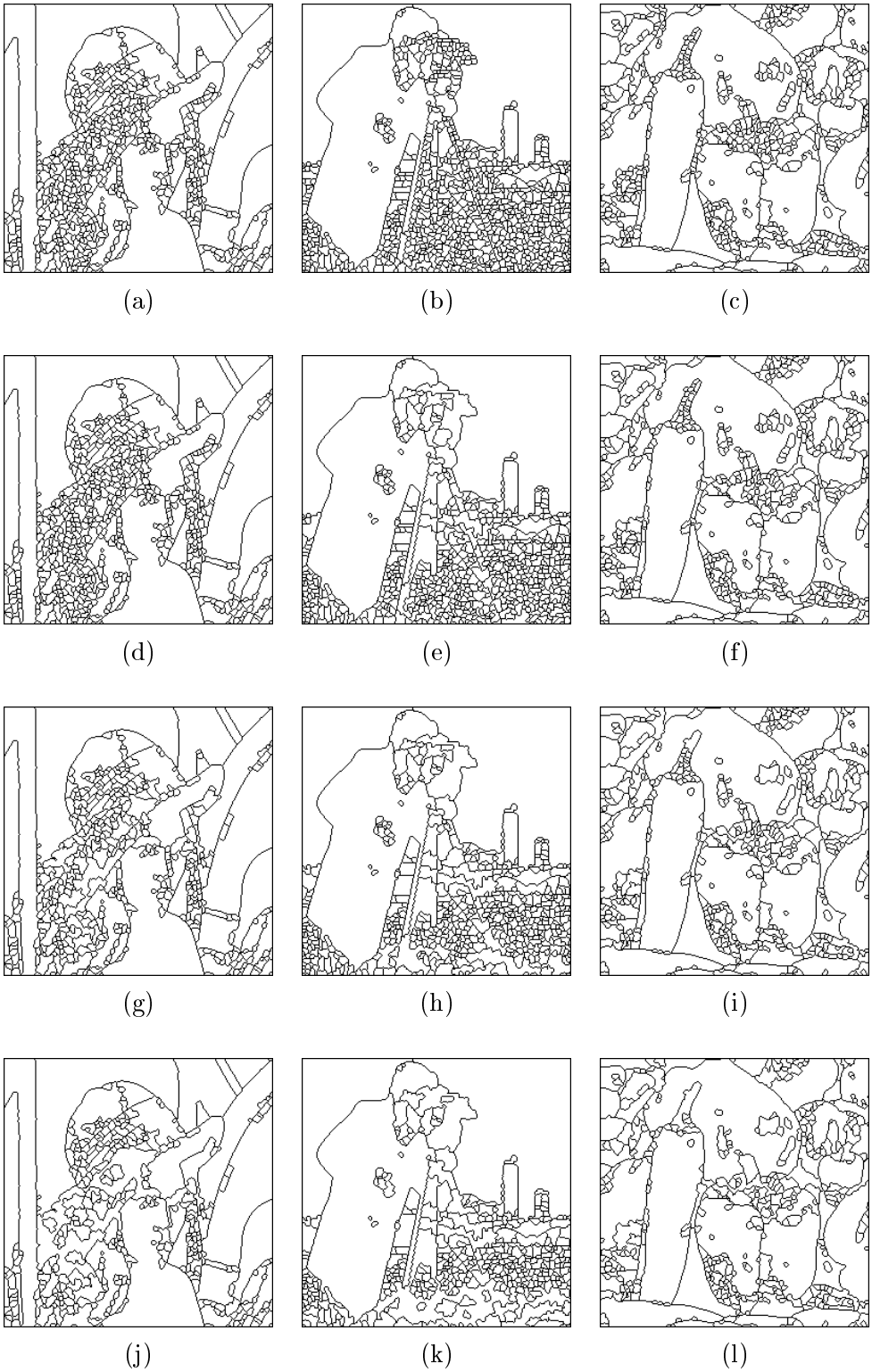
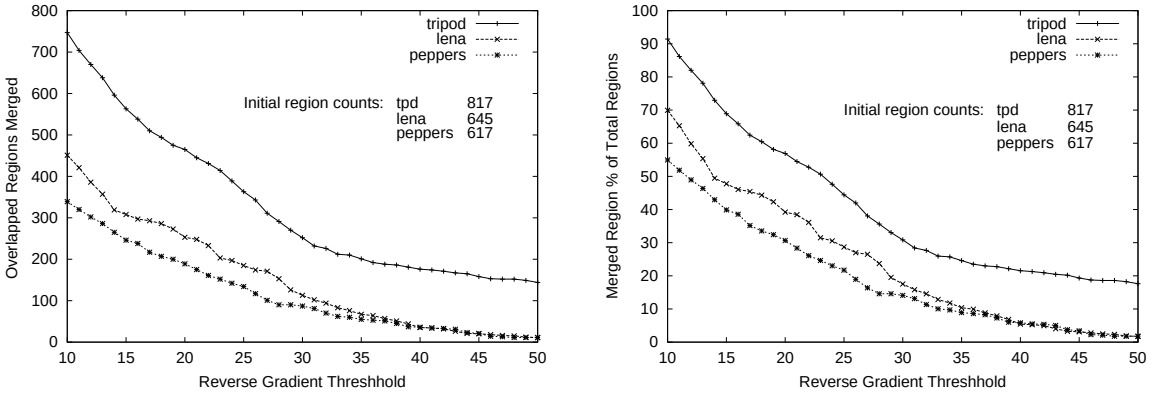


Figure 5: Results of drainage simulation region merging. The first row of images are border images prior of drainage simulations for the test images. The descending rows show the partitions after drainage simulation-based region merging with T_{∇} values of 32, 25, and 18.



(a) (b)

Figure 6: Results showing the affects of T_∇ on the partition. (a) shows the number of overlapped regions merged as a function of T_∇. (b) shows the number of overlapped regions merged as a percentage of the total regions in the image vs. T_∇.

the overall running time is slightly increased, the entire algorithm's complexity is not changed from $O(N)$. The region merging portion of the drainage simulation requires an examination of each region in the RAG. If a region is found to be overlapped, a subsequent examination of adjacent regions is required. This region merging is $O(V)$, where V is the number of regions in WS_X.

4 Conclusion

Drainage simulations employ a novel use of gradient image regional maxima in a WST application. The results show that the region count in the partitioned image is significantly reduced by utilizing drainage simulation-based region merging. These results are produced after an application of gradient magnitude thresholding using a relatively low threshold value. Drainage simulation-based region merging is constrained to spatially active areas of the image. The main benefit of this type of region merging is that it is done without requiring the examination or update of region descriptors. Applying this fast region merging initially reduces the overall region count before the more complex formal region merging is applied. The results likewise indicate that important edge information from the original image is not compromised by the drainage simulation-based region merging.

References

[1] P. Ausbeck. *Piecewise Smooth Modeling of Digital Images.* Ph.D. dissertation, Univ. California Santa Cruz, June 1996.

[2] S. Beucher. *Mathematical Morphology and Its Applications to Image Processing*, chapter Watershed, Hierarchical Segmentation and Waterfall Algorithm, pages 69–76. Kluwer, 1994.

[3] T. Blaffert, S. Dippel, M. Stahl, and R. Wiemker. The Laplace Integral for a Watershed Segmentation. In *International Conference on Image Processing*, volume 3, pages 444–447, September 2000.

[4] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*, chapter 23. McGraw-Hill Press, 1989.

[5] J.M. Gauch. Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies. *IEEE Transactions on Image Processing*, 8(1):69–79, January 1999.

[6] M. Grimaud. A New Measure of Contrast: Dynamics. In *Image Algebra and Morphological Processing III*, volume 2180, pages 292–305. SPIE, July 1991.

[7] M.W. Hansen and W.E. Higgins. Watershed-Driven Relaxation Labeling for Image Segmentation. In *Proceedings Eight International Conference on Image Processing*, pages 460–463, 1994.

[8] M.W. Hansen and W.E. Higgins. Watershed-based Maximum-Homogeneity Filtering. *IEEE Transactions on Image Processing*, 8(7):982–988, July 1999.

[9] 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.

[10] S.E. Hernandez and K.E. Barner. Joint Region Merging Criteria for Watershed-Based Image Segmentation. In *International Conference on Image Processing*, volume 2, pages 108–111, 2000.

[11] S. Makrogiannis, G. Economou, and S. Fotopoulos. Region Oriented Compression of Color Images Using Fuzzy Inference and Fast Merging. *Pattern Recognition*, 35(9):1807–1820, September 2002.

[12] Y.S. Moon and Tae Hyeon Kim. Efficient Morphological Segmentation Using a New Connected Operator. *Electronic Letters*, 1(36):22–24, January 2000.

[13] H. Sanderson and G. Crebbin. Image Segmentation for Compression of Images and Image Sequences. *IEE Proc. Vision and Image Signal Processing*, 142(1):15–21, February 1995.

[14] A.S. Wright and S.T. Acton. Watershed Pyramids for Edge Detection. In *International Conference on Image Processing*, volume 2, pages 578–581. IEEE, October 1997.