

# Análise de Algoritmos da Transformada *Watershed*

André Körbes , Roberto de Alencar Lotufo (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)

Faculdade de Engenharia Elétrica e de Computação (FEEC)

Universidade Estadual de Campinas (Unicamp)

Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{korbes,lotufo}@dca.fee.unicamp.br

**Abstract** – This paper presents an analysis of two issues of the watershed transform algorithms. The first issue regards the general behaviour of those, concerning the scanning order of pixels on the image, providing means for classification and generalization. The second issue is about the practical implications of the different definitions implemented by the algorithms in the literature. These are discussed on a real application of image segmentation.

**Keywords** – watershed transform, watershed algorithms

## 1. Introdução

A transformada *watershed* propõe uma abordagem morfológica para o problema de segmentação de imagens, interpretando estas como superfícies, onde cada *pixel* corresponde a uma posição e os níveis de cinza determinam as altitudes. A partir desta noção, deseja-se então identificar bacias hidrográficas, definidas por mínimos regionais e suas regiões de domínio. Intuitivamente, a transformada *watershed* trata de encontrar os pontos em uma superfície onde uma gota d'água possa escorrer para dois mínimos regionais diferentes. Esta analogia pode ser também criada inversamente, onde um nível d'água é elevado através de uma superfície, inundando a partir dos mínimos regionais, e, nos pontos onde águas provenientes de mínimos diferentes se tocarem ergue-se uma barreira, que constitui a linha de divisão das bacias.

Todavia, este trabalho se trata de uma análise dos algoritmos da transformada *watershed*, que implementam as diversas definições existentes. Para tal efeito, considerou-se a literatura a partir da introdução da primeira transformada rápida em 1991 por Vincent e Soille [10] até os trabalhos recentes de Cousty *et al.* [5], englobando 14 algoritmos. Assim, são identificadas características que permitem classificá-los e compara-se os resultados das definições destes em uma aplicação real.

Este trabalho é organizado conforme: a Sec. 2. apresenta as definições de transformada *watershed* de base para os algoritmos, a Sec. 3. propõe os métodos de exploração para classificação dos algoritmos, a Sec. 4. verifica a influências das definições em uma aplicação real, e a Sec. 5. apresenta as conclusões.

## 2. Definições de Watershed

Nesta seção introduzimos brevemente as principais definições de transformada *watershed*: inundação (Flooding-WT), distância topográfica (TD-WT), condição local (LC-WT), transformada imagem floresta com custo máximo de caminho (IFT-WT), zona de empate da IFT-WT (TZ-IFT-WT) e *watershed cut* (WC-WT). Estas definições são a base dos algoritmos disponíveis na literatura [6], e por consequência seus resultados afetam como as aplicações se comportam e o que pode ser esperado como saída destas.

A def. Flooding-WT consiste em um processo iterativo de limiarizações, onde as regiões crescem de acordo com suas zonas de influência calculadas em relação ao nível anterior. As linhas de *watershed* são os *pixels* que não pertencem a nenhuma região [10].

A def. TD-WT minimiza uma função de custo entre quaisquer pontos em uma imagem, baseada na soma das distâncias entre os pontos intermediários e relacionada às diferenças de níveis de cinza. Determina-se assim as regiões da imagem, provendo uma solução única onde um *pixel* com custo igual para dois ou mais mínimos regionais é marcado como *watershed* [9]. A def. LC-WT é derivada da def. TD-WT, onde a condição de unicidade da solução é removida. Desta forma, os *pixels* de *watershed* deixam de existir, sendo atribuídos a uma das regiões vizinhas [3].

A def. IFT-WT é uma abordagem baseada em floresta de caminhos mínimos para o problema, onde o custo do caminho é o máximo dos arcos. Todavia, em uma zona plana estes custos são iguais para todos os *pixels* nesta. Para tratar isto, é intro-

duzida uma segunda componente, o custo lexicográfico, que determina a menor distância até a borda da zona plana, desempatando a primeira [8]. A zona de empate da IFT-WT é uma transformada que unifica soluções. Dado que a IFT-WT produz um conjunto de soluções ótimas, a TZ-IFT-WT unifica-as em apenas uma solução, onde as regiões são definidas por aqueles *pixels* que tem um caminho ótimo para o mesmo conjunto de sementes em todas as soluções possíveis [1].

A def. WC-WT é uma transformada de corte de grafo, onde os cortes são feitos quando uma aresta satisfaz as condições de que os vértices que conecta levam a dois mínimos diferentes, e que seu valor é maior do que o valor de qualquer aresta nestes caminhos até os mínimos [5]. Entretanto, não é produzida uma solução única pois diversos cortes podem satisfazer estes critérios.

Uma consideração importante é que independentemente do algoritmo, seus resultados serão dependentes da definição implementada. Isto implica que algoritmos em uma mesma definição devem ser todos capazes de produzir o mesmo conjunto de soluções, ou um subconjunto deste.

### 3. Métodos de Exploração

A análise de exploração da imagem realizada neste trabalho busca caracterizar os algoritmos de acordo com a estratégia utilizada para a rotulação dos *pixels*. Entre as formas de exploração identificadas, há duas vertentes representativas e ainda outra alternativa pouco utilizada. Adotamos aqui a nomenclatura utilizada por Cormen *et al.* [4] e clássica na avaliação de algoritmos de busca em grafos: busca em largura e busca em profundidade. A terceira linha constitui-se da varredura aleatória, onde não é imposta nenhuma ordem no acesso aos *pixels*, pouco utilizada no âmbito da transformada por questões de desempenho. É importante ressaltar que esta caracterização não implica em qual definição o algoritmo implementa.

#### 3.1. Busca em Largura

Métodos de busca em largura são bem conhecidos na literatura de computação, sendo os fundamentos de diversos procedimentos, como os algoritmos de busca em grafos de Dijkstra para construção de SPFs e de Prim para construção de MSFs [4]. A principal característica deste tipo de algoritmo é

dada por sua natureza de expansão, sempre da última borda e uniformemente em sua largura. A respeito da distância da semente original, todos os vértices a distância  $k$  são visitados antes de visitar qualquer vértice a distância  $k + 1$  [4].

No campo dos algoritmos de transformada *watershed*, pode ser vista uma similaridade entre a busca em largura e o *watershed* por marcadores, onde um conjunto de sementes é expandido para se encontrar a partição ótima formada por estas, ou mesmo considerando-se como sementes os mínimos regionais. Estas propostas se diferenciam em uma série de características e implementam definições distintas, mas preservam a varredura em largura, onde a distância  $k$  passa a ser o custo do caminho.

Desta forma, pode-se generalizar os algoritmos que aplicam esta estratégia em 3 passos, onde permite-se variar a forma de expansão e rotulação de acordo com a definição adotada, com a variação fundamental sendo as conexões iterativas no passo 2:

1. Defina as sementes/marcadores
2. Calcule as conexões dos *pixels* da iteração atual com a anterior
3. Rotule os *pixels* de acordo com as conexões calculadas. Vá para o passo 2 e expanda as regiões da iteração atual, até visitar todos os *pixels*

De maneira geral, algoritmos de transformada *watershed* com varredura em largura tem como desvantagem a detecção inicial dos mínimos regionais, seja por um procedimento independente, ou por ordenação dos *pixels* por seus valores, para detecção dos componentes conexos mínimos. No entanto, se estes marcadores constituírem parâmetros de entrada do algoritmo, o processamento da vizinhança torna-se uma operação localizada e de baixo custo computacional.

#### 3.2. Busca em Profundidade

Assim como os algoritmos de busca em largura, a busca em profundidade também é muito comum na literatura de grafos, representada comumente por algoritmos gulosos. A estratégia adotada aqui é continuar a busca sempre a partir do vértice visitado mais recentemente, enquanto possível, e então retornar para analisar vértices pendentes. De fato, em

contraste com a busca em largura, a busca em profundidade prioriza vértices a distância  $k + 1$  assim que são descobertos, antes de visitar todos à distância  $k$  [4].

Pode-se facilmente ver a semelhança entre este procedimento descrito e uma gota d'água descedo sobre uma superfície, onde esta segue um caminho único. Em relação a transformada *watershed*, segue-se o mesmo raciocínio, representado pela técnica de *arrowing*, onde identifica-se para cada *pixel* o vizinho com menor valor. Diversos algoritmos utilizam esta abordagem, seguindo um caminho até que este seja esgotado em um mínimo regional, enquanto outros são baseados em construir os caminhos em etapas anteriores, identificar os mínimos, e por último executar a rotulagem percorrendo o caminho, caracterizando-os como buscas em profundidade.

Assim como o algoritmo de transformada *watershed* por busca em largura, pode-se generalizar a busca em profundidade em 3 passos, conforme a caracterização mencionada anteriormente:

1. Conecte cada *pixel* ao(s) seu(s) vizinho(s) de acordo com o custo desejado
2. Rotule os mínimos regionais
3. Percorra para cada *pixel* o caminho até o mínimo regional e rotule-o

Estes passos são geralmente distinguíveis nos algoritmos, no entanto podem ser unidos, na tentativa de atingir melhor desempenho. Em especial deve-se comentar sobre o passo 1, que depende da resolução das zonas planas. Estas regiões não podem, de maneira geral, ser conectadas utilizando um algoritmo de busca em profundidade, optando-se por utilizar variações de algoritmos em largura para propagação e divisão correta, aplicando-se uma segunda componente de custo para resolução destes empates. Entretanto, mesmo nestes casos, o passo 3 é característico de algoritmos de busca em profundidade, percorrendo o caminho até encontrar um *pixel* já rotulado ou um mínimo regional.

#### 4. Análise de Resultados

Nesta seção comparam-se os algoritmos de *watershed* por seus resultados em uma aplicação real. Desta forma, busca-se destacar a questão da definição implementada pelos algoritmos que pode impactar no resultado da aplicação. Considerando-se que cada um dos algoritmos está associado a

uma definição da transformada [6], serão apresentados apenas os resultados para cada uma destas. A def. Flooding-WT, que não é implementada corretamente por nenhum algoritmo, é representada aqui pelo algoritmo de Imersão de Vincent e Soille [10].

A aplicação apresentada aqui é um verificador de imagens de concreto em microscópio, onde deseja-se medir algumas regiões de interesse (ROI's), sendo a transformada *watershed* usada como um detector de texturas para regiões homogêneas, filtradas com base em um critério de área. A Fig. 1 apresenta as etapas para extração das ROI's: (a) imagem de entrada; (b) transformada *watershed* rotulada aplicada sobre o gradiente filtrado; (c) contornos internos das regiões obtidas após filtragem por área para selecionar regiões e remover ruído.

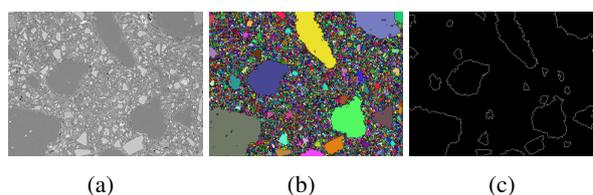
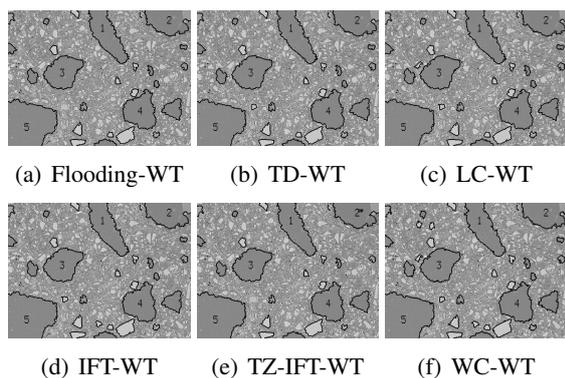


Figura 1. Etapas da aplicação *concrete*

Alternando a transformada *watershed* aplicada no passo (b) entre as seis definições, mede-se o número de regiões homogêneas detectadas, a área e variação para as cinco ROI's. A Fig. 2 apresenta as imagens produzidas pelos algoritmos, com o contorno interno destacado. A Fig. 2 (a)-(f) apresenta as cinco ROI's numeradas, e as soluções obtidas pelos algoritmos de Vincent e Soille [10], de Lin *et al.* [7], de Bieniek e Moga [3], de Beucher e Meyer [2,8], de Audigier, Lotufo e Couprie [1] e de Cousty *et al.* [5], respectivamente.

Conforme mencionado, o objetivo desta aplicação é medir regiões homogêneas, sendo que também há interesse no número de regiões detectadas. Para as imagens (a) - (f), foram encontradas 26, 27, 28, 28, 23 e 30 regiões respectivamente. A respeito das cinco ROI's, a área média e a máxima variação positiva (MVP) e negativa (MVN) foram calculadas, sendo apresentadas na Tab. 1, onde cada linha corresponde a uma ROI.

De forma geral, a análise dos resultados aponta para consistência entre as soluções. Entretanto, em alguns problemas uma análise mais aprofundada pode ser necessária, de modo a determinar



**Figura 2. Comparação de resultados da aplicação de watershed na identificação de regiões homogêneas**

Região	Média	MVP	MVN
1	12749,66	2,3%	3,5%
2	11343,5	1,9%	2,8%
3	11030,16	1,8%	2,9%
4	11041,83	6,2%	4,0%
5	22776	0,3%	1,0%

**Tabela 1. Medidas nas cinco ROI's da aplicação**

a abordagem mais adequada, especialmente no que diz respeito ao comportamento da definição, sendo que os algoritmos são dependentes destas, e pode ocorrer variação. Esta conclusão deve ser considerada em aplicações com baixa tolerância a erros. Todavia, quando há pré-processamento com grande simplificação da imagem, as diferenças tendem a ser pequenas, variando abaixo de 1%.

## 5. Conclusões

Neste trabalho foram analisados dois aspectos da transformada *watershed*: o comportamento de visitação da imagem dos algoritmos, classificados em largura e profundidade; e a influência das diferentes definições em uma aplicação real. Propusemos duas generalizações para os algoritmos de *watershed* de modo a compreender melhor o funcionamento destes, e permitir associar mais a transformada a princípios da computação como otimização de custos em caminhos e menos a princípios naturais como a gota d'água. A aplicação real foi testada com seis definições de modo a se verificar que, embora aproximados, os resultados variam e isso deve ser estudado caso a caso.

## Referências

- [1] R. Audigier, R. Lotufo, and M. Couprie. The tie-zone watershed: Definition, algorithm and applications. In *Proceedings of IEEE International Conference on Image Processing (ICIP'05)*, volume 2, pages 654–657, 2005.
- [2] S. Beucher and F. Meyer. *Mathematical morphology in image processing*, chapter The Morphological Approach to Segmentation: The Watershed Transformation. Optical Engineering. M. Dekker, New York, 1993.
- [3] A. Bieniek and A. Moga. An efficient watershed algorithm based on connected components. *Pattern Recognition*, 33(6):907–916, 2000.
- [4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 2 edition, 2001.
- [5] J. Cousty, G. Bertrand, L. Najman, and M. Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1362–1374, 2009.
- [6] A. Körbes and R. Lotufo. Analysis of the watershed algorithms based on the breadth-first and depth-first exploring methods. In *SIB-GRAPI'09*, pages 133–140, Rio de Janeiro, Brazil, Oct. 2009. IEEE Computer Society.
- [7] Y. Lin, Y. Tsai, Y. Hung, and Z. Shih. Comparison between immersion-based and toboggan-based watershed image segmentation. *IEEE Transactions on Image Processing*, 15(3):632–640, 2006.
- [8] R. Lotufo and A. Falcão. The ordered queue and the optimality of the watershed approaches. In *Proceedings of the 5th International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer Academic Publishers, June 2000.
- [9] F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.
- [10] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.