# A simpler approach to waterfall

Chris Nicholls, Irina Voiculescu and Stuart Golodetz

Oxford University Computing Laboratory

Most applications of image segmentations of Computerised Tomography (CT) scans require knowledge of where key anatomical features are found in the image.

@@@

- segmentation is - waterfall algorithm for segmentation - existing implementation (arguably overly complicated) - we have a better, simpler one - we will illustrate it on CT scans

Typically, the waterfall algorithm takes as input the output of another algorithm, the watershed, introduced by Beucher and Lantuejoul [2].

- say what the watershed does

@@@ A typical problem of the watershed algorithm is that it tends to over-segment images significantly, leading to far more regions than can be handled sensibly. This effect is due to problems such as indistinct boundaries between features, the variation present between different images, but mostly due to the algorithms not having any knowledge of the context in which the segmentation takes place.

due to - noise in the image - spurious local minima

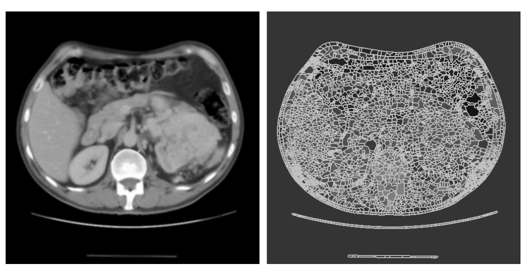

**Fig. 1.** Example of oversegmentation, output by applying the watershed algorithm to an axial slice of a CT volume. The individual regions are small and do not correspond to any anatomic features.

Figure 1 illustrates (on a CT scan) the results of a common segmentation algorithm, the watershed. The image is clearly over-segmented and hence not much more useful than the original for the purposes mentioned earlier.

The waterfall algorithm is an iterative process which can extract useful structure from this initial segmentation. The algorithm yields a partition forest hierarchy, which is a comprehensive data structure which can be used subsequently in the process of feature identification. Figure 2 illustrates the various layers that result from applying the waterfall algorithm to the segmentation shown in Figure 1. Each iteration of the algorithm yields a higher-level grouping of the regions in the previous layer.

Both the watershed and the waterfall algorithms are based on a geographical metaphor. The image is regarded as a landscape, with each grey value being proportional to the terrain height. The valleys are in the darker areas, whereas the lighter areas are regarded as peaks.

The waterfall algorithm [1, 3] can then be imagined as a flooding process. The water falls into (low) catchment basins and gradually fills them up to the nearest boundaries, sometimes

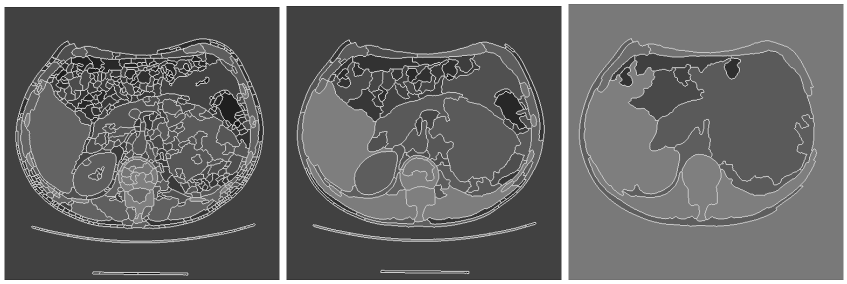

**Fig. 2.** Hierarchy of segmentations produced by applying the waterfall algorithm to the output of the watershed illustrated in Figure 1, showing regions merging successively.

spilling into adjacent regions. This process continues until the whole image becomes a single basin. The intermediate stages of the process can be regarded as intermediate segmentations of the image, with each basin representing a region.

The traditional implementation of this algorithm [3] involves the construction of a Minimum Spanning Tree (MST) and the gradual elision of some of its edges. Its nodes are initially the regions of the watershed output and its edges are the lowest pass points on the boundaries between these regions; the nodes and edges in subsequent layers are derived from these intitial ones through a merging process.

The collection of regional minimum edges of a graph $G$ is a connected subgraph of $G$ whose edges have the same weight as each other, and whose adjacent edges in $G$ have strictly higher weights. The waterfall algorithm relies heavily on finding these regional minimum edges, eliding them and rebuilding the MST – a process which not only requires careful implementation of the MST but, more importantly, is relatively complex and hard to implement.

In this paper we present a new data structure for the waterfall algorithm that simplifies the process and improves efficiency compared to current implementations. It is based on a recursive-tree data structure and a recursive relation on the nodes rather than the conventional iterative transformations.

The main advantage of our approach to the waterfall problem is that the algorithm uses a single loop to walk the MST and is therefore simpler to implement. For each iteration, it walks the MST bottom-up in a single pass and merges regions that belong together. The waterfall algorithm, thus improved, produces similar layers of segmented images, combined in a hierarchical structure that can be processed for feature identification.

A further advantage of our approach is that the algorithm can can be written in pure functional style. In particular, we have implemented it in Haskell. For this reason, the memory requirements are not directly comparable to existing imperative implementations, but we are about to integrate this new approach into an existing C++ code base.

We are also in the process of constructing a formal proof of correctness, which we hope to present at a later date. We have tested both algorithms on a number of small, measurable test cases and found that they produce the same output. Empirical tests indicate that this is also true of larger test cases, such as axial slices of CT volumes.

Production of partition forests in this manner also has many applications outside of the field of medical imaging, for instance, binary space partitioning in 3D map rendering for games.

## References


1. Serge Beucher. *Watershed, hierarchical segmentation and waterfall algorithm*. In Mathematical Morphology and its Applications to Image Processing, Proc. ISMM 94, pages 69-76, Fontainebleau, France, 1994. Kluwer Ac. Publ.
2. S. Beucher, C. Lantuejoul. *Use of watersheds in contour detection*. International Workshop on image processing, real-time edge and motion detection/estimation, Rennes, France, Sept. 1979.
3. Beatriz Marcotegui and Serge Beucher, *Fast Implementation of Waterfall Based on Graph*. In Mathematical Morphology: 40 Years On, Springer Netherlands, 2005.