Hierarchical watersheds with inter-pixel boundaries

No Author Given

No Institute Given

Abstract. Watersheds are the latest segmentation tool developed in mathematical morphology. These algorithms produce a segmentation of an image into a set of basins separated by watershed pixels. The over segmentation produced by these algorithms is reduced by removing all contours with a low saliency. This contour's saliency is generally defined from the minimal height of the watershed pixels along the contour. However, such a definition does not allow to define a contour's saliency in case of thick watersheds. Moreover, the set of basins which corresponds to the intuitive notion of regions do not define an image partition. In this paper we propose a method which allows to aggregate the watershed pixels to the basins while preserving the notion of contour and the associated saliency. The model used to encode the image partition is then decimated according to the contour saliency to obtain a hierarchy of partitions.

1 Introduction

Segmentation and contour extraction are important tasks in image analysis. Among the multitude of methods, the watershed transformation introduced by Lantuéjoul and Beucher [1] in the late 70's arises as a popular image segmentation algorithm. This method usually based on the gradient of the image presents the main advantage of providing closed curves leading to a proper definition of regions.

In mathematical morphology an image is considered as a topographic relief where each gray level (image intensity or image gradient) is interpreted as an altitude. The traditional watershed algorithm [12] simulates a flooding process. The minima of the image are pierced by holes and the water springing through the holes slowly immerse the whole relief. To prevent streams of water coming from different holes to intermingle, a dam is erected at the meeting locations. The set of all these obstacles represents the watersheds whereas the resulting separated lakes define the so-called basins attached to each regional minima.

Due to the flooding process one drip of water falling from a watershed pixel into an adjacent basin should follow an always descending path until the minimum of the basin. This property may be formalized as follows :

Definition 1 (Crest line property). *For each watershed pixel adjacent to a basin there exists an always descending path from the pixel to the regional minima of the basin.*

Note that not all watershed algorithms guarantee the preservation of this property [9].

A well known drawback of watershed algorithms is the over segmentation often produced by these algorithms. Since the contours appear to be correct the over segmentation problem turns out to be equivalent to a proper valuation of the saliency of each contour. One important feature to determine the contour saliency is its pass value generally defined [10] as the minimal height of the watershed pixels along the contour. This property may also be defined from the watershed pixel's pass values:

Definition 2 (watershed pixel's pass value). *Given a watershed pixel P adjacent to two basins, its pass value is defined as the minimal altitude one has to reach to connect the two basins while passing by P.*

Najman [10] proposed to valuate each contour by the minimal difference between the contour's pass value and the maximal depth of the two basins which merge along it during a flooding process. Such a valuation is called the dynamic of the contour.

However, the minimum of the pass values along a contour is quite sensitive to the noise that may appear along it. Moreover, despite their names, the watershed contours do not always define valid connected paths. Indeed, Najman and Vincent [9, 12] have shown that the existence of thick watershed areas is induced by the definition of the watersheds and can't thus be avoided. The determination of the adjacency between the basins and the valuation of the associated contours is then conditioned to a proper thinning [8] of the thick watershed areas.

In this paper we propose a method which allows to aggregate the watershed pixels to the basins while preserving the pixel's pass value information (Section 2). The contours between the basins are encoded by a model based on inter pixel boundary paths, the pixel's pass values being stored within this model (Section 3). The different paths encoding the borders of the partition are then encoded by a graph data structure and a decimation process based on the dynamic of contours is applied on the graph in order to obtain a hierarchy of image partitions (Section 4).

2 Computing the altitude of pixel watershed

As mentioned in Section 1, a watershed algorithm produces a partition of an image into a set of basins $B_1, \ldots, B_p$ and a set of watershed pixels $\mathcal{W}$. In the following of this article we will consider that the 4 neighborhood is used for the basins and that all watershed pixels adjacent to a basin satisfy the crest line property (Definition 1).

The saliency of a contour between two basins is generally measured from the contour's pass value [10] (Section 1). However, the definition of a watershed pixel's pass value (Definition 2) does not hold for thick watershed areas where many watershed pixels are adjacent to a single basin or surrounded by other watershed pixels and thus not adjacent to any basins.

Thick watersheds induce two different problems for the determination of the contour's pass values. First of all, within the watershed framework a contour corresponds to a border between two basins. In case of a thick watershed area, the basic idea which consists to consider all basins adjacent to the thick area as adjacent may not lead to a valid partition of the image into 4 connected regions. Therefore, the adjacency between the basins and thus the existence and location of the contours is relative to a labeling of the watershed pixels to the different basins. Secondly, in order to be coherent with Definition 2, the definition of the pass value on the resulting contours should correspond to the minimal altitude one has to climb to connect the two basins separated by such a contour.

We perform the labeling of watershed pixels by an aggregation process which merges all watershed pixels into a set of final basins $B'_1, \ldots, B'_p$. Let us define the altitude of a watershed pixel $P \in B'_i$ as the minimal height one as to reach from the minimum of B'_i to reach P. Note that the altitude of a watershed pixel corresponds to its pass value when the pixel is adjacent to at least two basins.

The computation of the watershed pixel's altitude requires to consider all paths joining a watershed pixel $P \in B'_i$ to the connected component m_i of the basin B_i with a minimal height. This set of paths is defined as follows :

$$\Pi_i(P) = \{\pi \subset B'_i | \pi(1) \in m_i \text{ and } \pi(q) = P\} \text{ with } q = |\pi| \tag{1}$$

The altitude of P is then defined as :

$$Alt(P) = min_{\pi \in \Pi_i(P)} max_{j \in \{1,\ldots,|\pi|\}} h(\pi(j)) \tag{2}$$

where h denotes the height of each pixel in the image.

Note that, using the crest line property, the minimum m_i of B_i may be replaced by B_i in equation 1 without changing the value of $Alt(P)$.

Using the above definitions, the altitude of a watershed pixel is equal to its height only if there is an always descending path included in the basin to which it is aggregated and which connects it to the minimum of the basin. Moreover, since a local minima defines a basin, a thick watershed area can not contain any local minima. Therefore, any watershed pixel in a thick watershed area can be connected to a basin by an always descending path. The determination of such paths is insured by Algorithm 1 which performs an immersion process on the watersheds (Fig. 1). More precisely, this algorithm performs the following steps:

1. All watershed pixels adjacent to an initial basin B_i are put into a queue,
2. while the queue is not empty
   (a) One pixel with a minimal altitude is removed from the queue and merged with an adjacent basin.
   (b) All the watershed pixels adjacent to the removed pixel and not already in the queue are added to it.

Each watershed pixel is either adjacent to a basin or put into the queue by an adjacent watershed pixel. This adjacent watershed pixel being merged into a basin, each pixel removed from the queue is adjacent to at least one basin.

```
input : watersheds 𝒲, bassins {B_1,...,B_p}, image h
ℬ = ⋃_{k=1}^{p} B_k
ℱ_0 : set of watershed pixels adjacent to a bassin
N(P): 4 neighborhood of P.
k=0
while ℱ_k ≠ ∅
     select P_{k+1} such that h(P_{k+1}) = min{h(P), P ∈ ℱ_k}
     ℱ_{k+1} = (ℱ_k − {P_{k+1}}) ∪ [N(P_{k+1}) ∩ (𝒲 − ℱ_k)]
     select j such that N(P_{k+1}) ∩ B_j^{(k)} ≠ ∅
     B_j^{(k+1)} = B_j^{(k)} ∪ {P_{k+1}},  ∀i ≠ j  B_i^{(k+1)} = B_i^k
     𝒲 = 𝒲 − {P_{k+1}}
     k=k+1
```

Algorithm 1: *Immersion of watershed pixels*

Moreover, the algorithm removes at each step one pixel from the watershed set. This set being finite the algorithm terminates. Finally, once one watershed pixel is processed all its neighbors must be processed since they are put into the queue which is empty when the algorithm terminates. Therefore, if we suppose that one watershed pixel is not processed by the algorithm we must suppose that the whole connected component of watershed pixels including it is not processed. This last assumption contradict the fact that the image is partitioned into the set of watersheds and initial basins. Indeed, all pixels adjacent to a basin are initially put into the queue and thus processed. Therefore the algorithm terminates and proceeds all watershed pixels.

The proof that we can find for each watershed pixel P an always descending path included in the final basin to which P is aggregated and connecting P to the minimum of this basin may be established using a recurrence hypothesis. The basic idea of the proof is as follows: If we suppose that at step k all previously dequeued pixels satisfy the property, the pixel P_{k+1} should have a greater height than these pixels (otherwise it would have been dequeued before them). Therefore, P_{k+1} is either directly adjacent to a basin and the proof is provided by the crest line property (Definition 1) or it is surrounded by watershed pixels and adjacent to a previously dequeued one. In this case we can concatenate P_{k+1} to the path associated to one of its dequeued neighbor. The resulting path is always descending by construction.

The queue may be efficiently implemented by an array of lists. In this case each watershed pixel is considered twice: once to be put into the queue and once to be removed from it. Note that at each step of Algorithm 1, several pixels with a minimal altitude may belong to the queue. The order defined on such pixels influences the growing speed of the different basins and thus the final partition. A priority may be defined on these pixels based either on a geodesic distance between the basins or an external criteria such as a distance between the feature vectors of the watershed pixels and the ones of the basins. Note however, that the

2	3	6	2
3	6	6	6
255	7	6	6
2	255	7	6
1	2	255	5

(a) initial image

2	3	6	2
3	6	6	6
255	7	6	6
2	255	7	6
1	2	255	5

(b) watersheds

A_2	A_3	A_6	A_2
A_3	A_6	A_6	B_6
C_{255}	A_7	B_6	B_6
C_2	C_{255}	D_7	D_6
C_1	C_2	C_{255}	D_5

(c) partition

Fig. 1. A thick watershed(b) produced by the initial image (a) and the resulting image partition produced by Algorithm1(c). The indexes in (c) denote the height of the pixels.

priority that may be established on the queue has no influence on the altitudes of the watershed pixels along the final contours.

3 Transferring pixel's altitudes to lignels

Algorithm 1(Section 2) aggregates each watershed pixel to a basin and provides thus a partition of the image into a set of final basins $B'_1, \ldots, B'_p$. One may think to define the border between two basins B'_1 and B'_2 as the set of watersheds belonging to B'_1 (respectively B'_2) and having one neighbor in B'_2 (respectively B'_1). However, assuming a 4 connectedness for the basins, such a definition of borders does not defines valid 8-connected contours between regions. Indeed, within thick areas two adjacent watershed pixels may belong to different basins. The border between both basins is in this case two pixel large.

This last drawback may be overcome by defining basins's boundaries as *inter-pixel boundary paths* [6, 5]. Using such a representation, the boundaries of the basins are defined as 4-connected paths in the $P_{\frac{1}{2}}$ plane (Figure 2(b)):

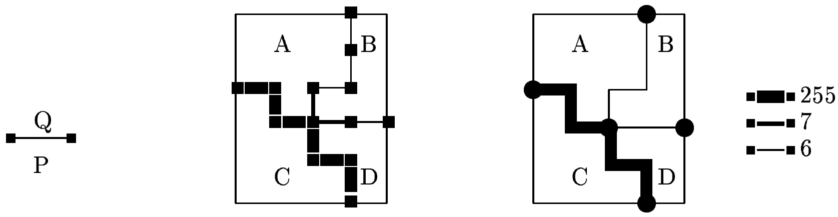
$$P_{\frac{1}{2}} = \{(i + \tfrac{1}{2}, j + \tfrac{1}{2}), \text{ with } (i, j) \in \mathbb{Z}^2\}.$$

This approach has been first described by Brice and Fennema [2] when introducing grouping segmentation algorithms. Later several discrete topologies have been developed which provide formal tools to study such a representation [6, 5].

Two adjacent successive half integer points along an inter-pixel boundary path are said to be joined by a *lignel* [5] (also denoted crack [11]). Each lignel joins two half integer points along a boundary path between two basins and separates two pixels belonging to these two basins (Figure 2(a) and (b)). The set of lignels encoding the borders of the partition is denoted by $\mathcal{L}$.

The definition of the watershed pixel's pass value (Definition 2) may thus be extended to lignels as follows :

Definition 3 (Lignel's pass value). *Given one lignel $l \in \mathcal{L}$ between two pixels P and Q belonging to two different basins, B'_i, B'_j the pass value of l is defined as the minimal altitude one as to reach to connect B'_i and B'_j while passing by P and Q.*



(a) one lignel (b) Inter pixel boundaries (c) nodes and segments (d) pass values

Fig. 2. One lignel element (a) and the encoding of the image partition (Fig. 1(b)) by inter pixel boundary paths(b). The symbols ■ denote the half integer points encoding the borders of the partition. The width of the lines in (d) represent the values attached to lignels(b) and segments(c).

Given two basins, B'_1 and B'_2 and a lignel l between two watershed pixels P and Q belonging respectively to B'_1 and B'_2, the pass value of l may be formally defined using the set of paths joining B'_1 and B'_2 and passing by P and Q :

$$\begin{aligned} \Pi(P,Q) = \quad & \{\pi \subset B'_1 \cup B'_2 | \pi(0) \in B_1, \pi(q) \in B_2 \text{ and} \\ & \exists j \in \{1,\ldots,q-1\} \text{ such that } (\pi(j),\pi(j+1)) = (P,Q)\} \end{aligned}$$
with $q = |\pi|$.

The pass value of l is then defined as in Section 2 by:

$$pass\_value(l) = min_{\pi \in \Pi(P,Q)} max_{j \in \{1,\ldots,|\pi|\}} h(\pi(j))$$

Let us suppose that P and Q are both watershed pixels. Using the image partition provided by Algorithm 1, there is two always descending path π_1 and π_2 respectively from P to B'_1 and from Q to B'_2. If we denote by π_2^{-1} the reverse path of π_2, the path $\pi = \pi_1.\pi_2^{-1}$ belongs to $\Pi(P,Q)$ and has an altitude equal to $max(h(P), h(Q))$. Since all paths in $\Pi(P,Q)$ must pass by P and Q, π has a minimal altitude. If P (resp. Q) is not a watershed pixel we have by the crest line property (Definition 1) $h(Q) > h(P)$ (resp. $h(P) > h(Q)$) and in this case the pass value of l is equal to the height of Q (resp. P). Therefore, in all cases the pass value of $l \in \mathcal{L}$ is equal to the maximal height of P and Q:

$$\forall l \in \mathcal{L} \quad pass\_value(l) = max(h(P), h(Q))$$

where l separates the pixels P and Q.

4 Computation of contour's dynamics

The methods described in Sections 2 and 3 allows us to encode the boundaries of the partition by a set of inter-pixel boundaries points joined by lignel elements. Each lignel is valuated by its pass value (Definition 3).

This set of lignels may be structured as follows: We call a *segment* a maximal path between two basins (Figure 2(c)). The intersection between several segments is called a *node*. Moreover, in order to preserve the topology consistency it is necessary to add an arbitrary node on each boundary reduced to a single loop. It can be shown [4] that a closed single path in the $P_{\frac{1}{2}}$ plane satisfies Jordan's theorem [6]. Therefore, any connected path joining two pixels respectively inside and outside a basin must cross the basin's boundary. This last property allows us to define implicitly each basin by a closed segment or by the concatenation of the segments and nodes which belong to its boundary.

Each segment is thus composed of a sequence of valuated lignels encoding the different pass values along the segment. The pass value of each segment should thus be defined from the ones of its lignels. In our implementation, the segment's pass value is fixed to the median value of its lignel's pass values. Note that this value is generally fixed to the minimal altitude along the contour which roughly corresponds in our model to the minimal pass value of the lignels along the segment. Our choice of a median value prevents the pass value of the contour to be artificially lowered by the presence of noise along the contour.

The set of nodes and segments can then be encoded with a graph by associating nodes and segments respectively with vertices and edges. The pass value of each segment is thus attached to the associated edge. The basins are then defined as the faces of the graph and may be encoded by the vertices of the dual graph. The initial graph and its dual may be encoded using either the dual graph [7] or combinatorial map [3] models. We choose to use the combinatorial map model which allows us to encode only implicitly the structure of the dual graph.

Given the edge's pass values, the dynamic of the edges are computed [10] (Section 1). A hierarchy of segmentations is then built on the edge's dynamics by removing iteratively the edges with the lowest dynamics.

Figure 3 shows the dynamic of contours (b) computed on the Lenna test Image(a) using a Deriche gradient operator as initial image for the watersheds algorithm [12]. The partition obtained by removing all edges with a dynamic lower than 16 is shown in Figure 3(c).

5 Conclusion

We have defined in this article a construction scheme for hierarchical watersheds based on inter-pixel boundaries. This representation allows to remove the ambiguities induced by the presence of thick watershed areas without loss of information. Moreover, our model may be used in conjunction with any watershed algorithm satisfying the crest line property. Our model encodes a partition of the image into 4-connected basin. The topological soundness of this model and its theoretical background allows us to encode it efficiently using combinatorial maps. Our future work will consist to enhance the definition of the segment's pass value based on the values of its lignels. Such results have a direct influence on the value of the dynamics and thus on the final hierarchy.

(a) initial image (b) segment's dynamics (c) dynamics's thresholding

Fig. 3. The dynamic of the segments (b) deduced from the watersheds of image (a) and one level of the hierarchy where all contours with a dynamic lower than 16 have been removed(c)

References

1. S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing*, Rennes, 1979. CCETT/IRISA.
2. R. Brice and C. Fennema. Scene analysis using regions. *Artificial intelligence*, 1:205–226, 1970.
3. L. Brun and W. Kropatsch. Combinatorial pyramids. In Suvisoft, editor, *IEEE International conference on Image Processing (ICIP)*, volume II, pages 33–37, Barcelona, September 2003. IEEE.
4. J. P. Domenger. *Conception et implémentation du noyau graphique d'un environnement* $2D\frac{1}{2}$ *d'édition d'images discrètes*. PhD thesis, Labri Université Bordeaux I, 351 cours de la libération 33405 Talence, avril 1992.
5. J. Françon. Topologie de Khalimsky et Kovalevsky et algorithmes graphiques. In *First Colloquium on Discrete Geometry in Computer Imagery*, Strasbourg, September 1991.
6. E. Khalimsky, R. Kopperman, and P. Meyer. Boundaries in digital planes. *Journal of applied Math. and Stocastic Analysis*, 3:27–55, 1990.
7. W. G. Kropatsch and H. Macho. Finding the structure of connected components using dual irregular pyramids. In *Cinquième Colloque DGCI*, pages 147–158. LLAIC1, Université d'Auvergne, ISBN 2-87663-040-0, September 1995.
8. J. Marchadier, D. Arquès, and S. Michelin. Thinning grayscale well-composed images. *Pattern Recognition Letters*, 25:581–590, 2004.
9. L. Najman and M. Couprie. Watershed algorithms and contrast preservation. In *Discrete geometry for computer imagery*, volume 2886, pages 62–71. LNCS, Springer Verlag, 2003.
10. L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEETPAMI*, 18(2):1163–1173, December 1996.
11. A. Rosenfeld. Digital topology. *Amer. math. monthly*, 86:621–630, 1979.
12. 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.