

Ultimate Opening and Gradual Transitions

Beatriz Marcotegui¹, Jorge Hernández¹, and Thomas Retornaz²

¹ MINES ParisTech, CMM- Centre de Morphologie Mathématique,
Mathématiques et Systèmes, 35 rue Saint Honoré - Fontainebleau, France
`beatriz.marcotegui@mines-paristech.fr`

`http://cmm.ensmp.fr`

² A2iA SA, Artificial Intelligence and Image Analysis
40 bis rue Fabert, 75007 Paris, France
`thomas.retornaz@a2ia.com`

Abstract. Ultimate Opening (UO) is a powerful operator based on numerical residues. In a multi-scale framework, it analyzes an image under a series of increasing openings. Contrasted objects are detected when they are filtered out by an opening, generating an important residue. Gradual transitions make this operator underestimate the contrast of blurred objects. In this paper we propose a solution to this problem, integrating series of non-null residues. The resulting operator handles correctly blurred boundaries, without modifying the behavior on sharp transitions.

Keywords: Numerical residues, ultimate opening, attribute opening, image segmentation, mathematical morphology.

1 Introduction

Given an image I , a residual operator ($r(I)$) in Mathematical Morphology is defined as the difference between two operators (Ψ and ζ) applied to the given image I : $r(I) = \Psi(I) - \zeta(I)$. Morphological gradient, defined as the residue of dilation and erosion ($\delta(I) - \epsilon(I)$), or top-hats, defined as the residue of the image and its opening ($I - \gamma(I)$) or its closing ($\phi(I) - I$) are residual operators extensively used. The extension of residues to a family of primitives Ψ_i and ζ_i is defined as the union of residues computed at different sizes i : $r(I) = \cup_i (\Psi_i(I) - \zeta_i(I))$. Ultimate erosion is an example of this type of operators. It is defined as the union of residues of erosions ($\Psi_i = \epsilon_i$) and openings by reconstruction ($\zeta_i = \gamma_i^{rec}$): $UltimateErosion(I) = \cup_i [\epsilon_i(I) - \gamma_i^{rec}(I)]$. Ultimate erosion is used for segmenting binary connected objects. Another example is the skeleton: the union of residues of erosions and the corresponding openings $Skeleton(I) = \cup_i [\epsilon_i(I) - \gamma_i(I)]$. The application of ultimate erosion or skeleton to binary images produces only one non-null residue (of index i) for each pixel. The union of these residues is then straightforward. This is not the case for gray level images. Thus, the extension of residual operators with a family of primitives to gray level images has to deal with the combination of several residues at different sizes for the same pixel. Beucher in [1] proposes to keep for

each pixel the maximum residue and the index i corresponding to the size leading to it. Thus, he defines the quasi-distance function as the maximum residue between consecutive erosions ($QuasiDistance(I) = \max_i (\epsilon_i(I) - \epsilon_{i+1}(I))$) and the ultimate opening (UO) as the maximum residue between consecutive openings ($UltimateOpening(I) = \max (\gamma_i(I) - \gamma_{i+1}(I))$). Retornaz and Marcotegui in [2] introduce ultimate attribute openings, based on attribute operators [3], and Fabrizio [4] proposes an efficient implementation based on a max-tree representation. Leite and Guimarães propose an image filtering framework based on residues by attribute, but requires a set of parameters (size or complexity) in order to select which regions will be preserved [5].

(UO) is a powerful residual operator, able to segment generic images without a priori size information. It has been successfully used for several applications: granulometry of rocks [6], automatic text localization [2] and façade segmentation [7]. In this paper we focus on the behavior of this operator on gradual transitions and propose a solution to avoid underestimating the contrast of blurred objects. This paper is organized as follows: section 2 reminds the ultimate opening principle, section 3 analyzes the problem of gradual transitions and proposes a solution to deal with blurred objects, section A explains the implementation of the new operator, section 4 illustrates the performance of the new operator and finally, section 5 concludes the paper.

2 Ultimate Opening

2.1 Ultimate Opening Definition

Following a multi-scale approach, UO analyzes the image while performing a series of openings of increasing sizes: $\gamma_\lambda(I)$, with $\lambda = \{0, 1, \dots, N - 1\}$ and N the maximum opening size considered. The series of differences between consecutive openings, named residues ($r_\lambda(I) = \gamma_\lambda(I) - \gamma_{\lambda+1}(I)$), is computed. Each pixel keeps two significant pieces of information:

- the maximal residue, $R_\theta(I) = \max_\lambda(r_\lambda(I))$. It is the strongest change generated by an opening. An important structure is supposed to be filtered out by this opening and the corresponding residue estimates its contrast.
- the size of the opening leading to the maximal residue $q_\theta(I)$. If several openings lead to the same maximum $r_\lambda(I) = R_\theta(I)$, the largest λ among them is chosen (as proposed in the UO definition [1]). $q_\theta(I)$ is set at 0 for pixels where all residues are null. This happens for the minimum of the image or for objects larger than N , the largest opening size considered.

The definition of the UO is then written as:

Definition 1 (Ultimate opening [1]). *The ultimate opening operator, θ , of an image I is given by:*

$$\theta(I) : I \xrightarrow{\theta} (R_\theta(I), q_\theta(I)) \quad (1)$$

where,

$$R_\theta(I) = \max_\lambda(r_\lambda(I)) = \max_\lambda(\gamma_\lambda(I) - \gamma_{\lambda+1}(I)) \quad (2)$$

$$q_{\theta}(I) = \begin{cases} \max \{ \lambda + 1 \mid r_{\lambda}(I) = R_{\theta}(I) \} & R_{\theta}(I) > 0 \\ 0 & R_{\theta}(I) = 0 \end{cases} \quad (3)$$

where, $(\gamma_{\lambda})_{\lambda \in \{0,1,\dots,N\}}$ is an increasing family of openings.

Labeling $q_{\theta}(I)$ we get a partial partition [8,9]: pixels are grouped in non intersecting segments that do not cover the whole domain ($q_{\theta}(I)$ is set at 0 for pixels where all residues are null).

Ultimate opening segments light objects in a dark background. In order to deal with dark objects in a light background, an ultimate closing should be used. The ultimate closing can also be applied to the gradient image, dealing with both polarities at the same time. The use of a gradient image is not recommended for images containing thin objects.

In this paper we focus on attribute openings. If not specified in the text, height attribute (y -extent of connected components) will be used in the following.

2.2 Example of Application

Fig. 1 shows the ultimate height closing of a real image 1(a). The letters of the image are not homogeneous. They are filled with a texture from another image. Fig. 1(b) shows $q_{\theta}(I)$ and Fig. 1(c) $R_{\theta}(I)$. In spite of the fact that the letters



(a) Original Image



(b) $q_{\theta}(I)$ (size information)

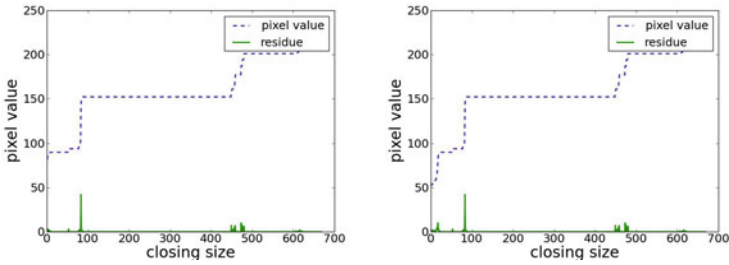


(c) $R_{\theta}(I)$ (contrast information)

Fig. 1. Ultimate Height Closing : (b) $q_{\theta}(I)$, represented in false colors in order to see the segmented regions and (c) $R_{\theta}(I)$

are textured, most of them are correctly segmented, because their texture is less contrasted than the contrast between the characters and their background.

In order to understand this process, let us see the evolution with the series of closings (characters are darker than their background) of two pixels inside the same letter. Two pixels inside the first letter “N” of “GENERATION” are randomly chosen. The series of values of these two pixels, after applying closings of increasing sizes, are shown in blue dashed line in Fig. 2(a) and 2(b). The corresponding residues (the derivative of the blue dashed curve) are illustrated in green solid curves of Fig. 2(a) and Fig. 2(b). Residues for small closing sizes correspond to the internal fluctuations of the letter. These residues are different for different pixels of the same letter. However, both pixels have their highest residue for a closing of size 83. This is because the letter is 82 pixels high, and it is filtered out by a closing of size 83 (see Fig. 2(c) and Fig. 2(d)). The same important residue is seen by all the pixels of the structure. Thus, ultimate closing catches contrasted structures when they are filtered out, leading to interesting segmentation results without a priori information.



(a) Pixel evolution (in blue dashed line) and corresponding residues (in green solid line) of a pixel inside “N”.

(b) Pixel evolution (in blue dashed line) and corresponding residues (in green solid line) of another pixel inside the same “N”



(c) Closing of size 82



(d) Closing of size 83

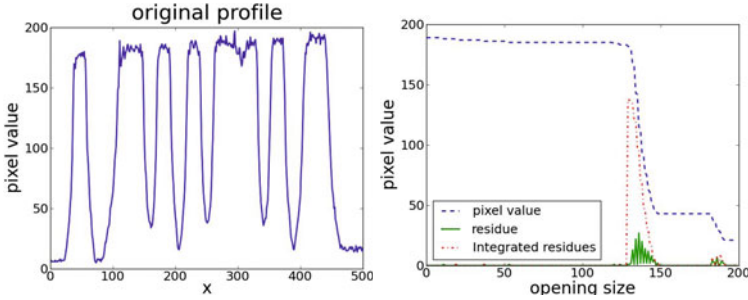
Fig. 2. Intermediate images of Ultimate Height Closing

3 Gradual Transitions and Δ Ultimate Opening

Ultimate opening produces interesting segmentation results in very diverse situations. Nevertheless, it underestimates the contrast of blurred objects. Indeed, the



(a) Original image



(b) Original pixel values of white line in (a) (c) Pixel value, residues and integrated residues of a pixel inside letter “e”.

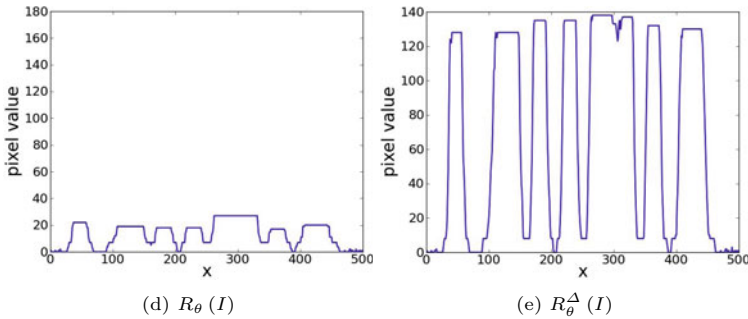


Fig. 3. Residues of a gradual transition (blurred objects)

boundary of a blurred object is a gradual transition. The contrast of the object is then divided into several steps. The contrast associated by UO to the object would be the largest of these steps, inevitably smaller than the real contrast of the object. An example of this situation is shown in Fig. 3. Fig. 3(a) shows the original image and Fig. 3(b) the profile of a horizontal line (superimposed in white in Fig. 3(a)). The gray level value of the letters is about 160 while the background value is about 10. The contrast of letters is then about 150. But the contrast estimated by the UO is only about 20 (see R_θ in Fig. 3(d)). Fig. 3(c) shows the pixel value evolution (in blue dashed line) of a pixel inside a letter, with a series of increasing openings. The corresponding residues are shown in green solid line. We can observe a series of non null residues for consecutive opening sizes, corresponding to the gradual transitions of the blurred boundary. In fact, transition regions are characterized by their small size. This is why small residues appear for consecutive opening sizes. If we integrate the series of non

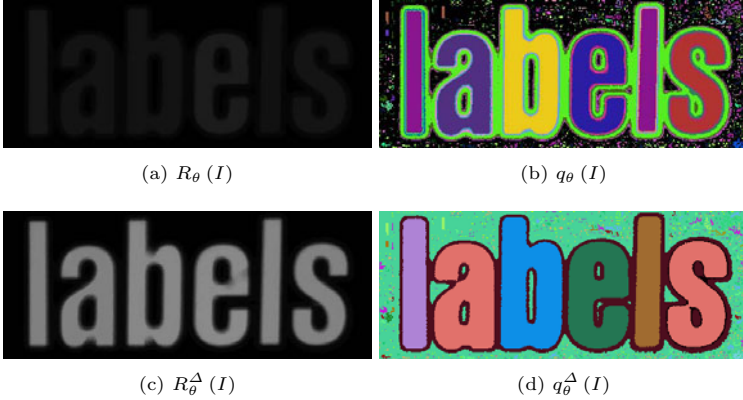


Fig. 4. *UAO* and ΔUAO comparison



Fig. 5. Transition zones, detected as those that benefit from the residues integration

null residues, assuming that they correspond to transition zones, we get a much better contrast estimation. Red dashdotted line in Fig. 3(c) shows the integrated residues. Fig. 5 shows which pixels benefit from the integration. We can see that they are located in the boundary of the blurred objects.

This idea has been generalized to introduce the ΔUO , that integrates the series of residues until finding a series of Δ null residues. For $\Delta = 0$ we get the classic *UO*, $\Delta = 1$ integrates series of non-null residues, $\Delta = 2$ integrates series of residues separated by at least 2 consecutive null residues, and so on. The larger Δ is, the larger transition zones can be.

Fig. 4(a) and Fig. 4(b) show R_θ and q_θ of 3(a) while Fig. 4(c) and Fig. 4(d) show R_θ^Δ and q_θ^Δ with height attribute openings, and $\Delta = 1$. We can appreciate a much better contrast estimation of ΔUAO .

4 Results

In this section we report several results, illustrating the performance of ΔUAO with $\Delta = 1$ in several complex images from the public ICDAR 2003 image database [10]. In order to deal with both polarities ΔUAO is applied to image I and to its inverse I^c . For each pixel, the polarity leading to a bigger residue is kept (see Eq.4).

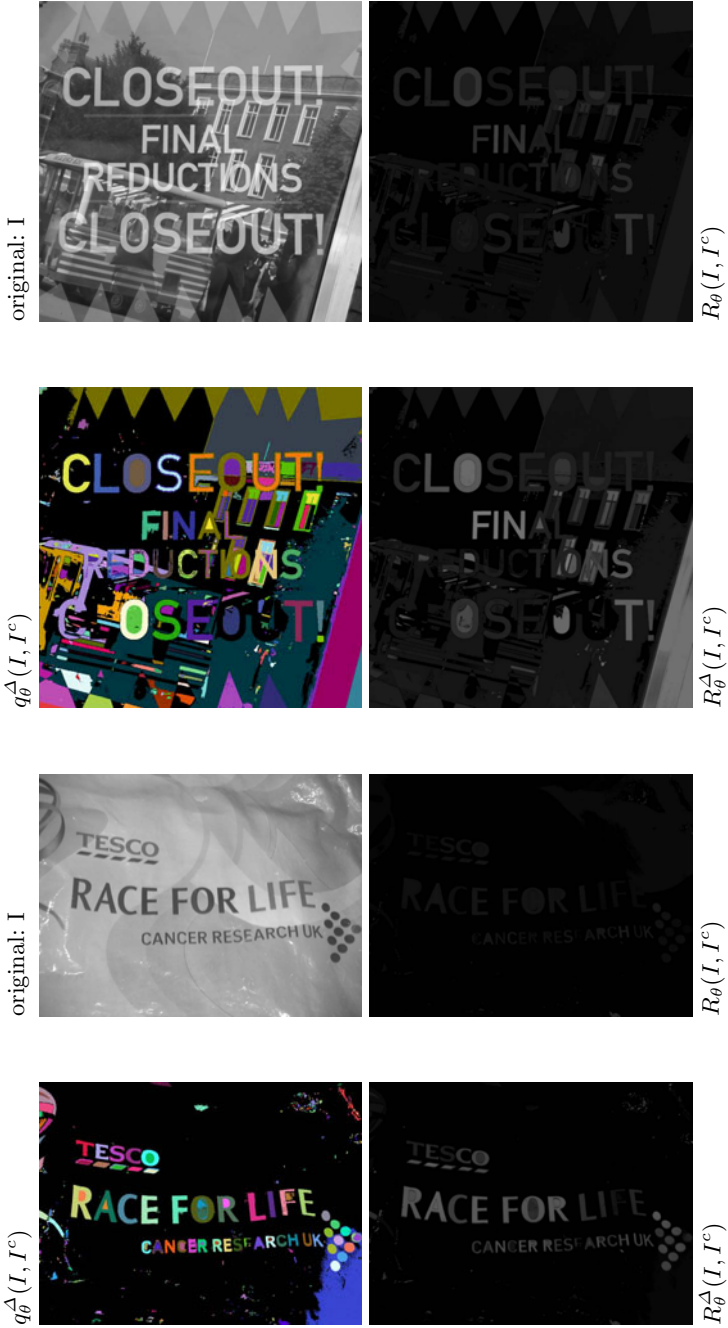


Fig. 6. ΔUAO Results of Δ ultimate height opening. For each image, first row: original image and R_θ ; second row: $q_\theta^\Delta(I, I^c)$ and $R_\theta^\Delta(I, I^c)$.

$$\begin{aligned}
 R_\theta(I, I^c) &= \max(R_\theta(I), R_\theta(I^c)) \\
 q_\theta(I, I^c) &= \begin{cases} q_\theta(I) & R_\theta(I) > R_\theta(I^c) \\ q_\theta(I^c) & \text{otherwise} \end{cases}
 \end{aligned} \tag{4}$$

The gradient is not used in order to be able to detect thin objects.

Fig. 6 compares classical *UAO* with Δ *UAO*. For each image, the first row shows the original image (on the left) and the classical *UAO* residue, according to equation 4. The contrast of the selected structures is clearly underestimated. In the second row, $q_\theta^\Delta(I, I^c)$ (on the left), represented in false colors in order to see the segmented regions and its corresponding residual information $R_\theta^\Delta(I, I^c)$ (on the right). Satisfactory results have been obtained in difficult situations: complex background or illumination problems. Most letters have been correctly segmented in q_θ^Δ and their associated contrast in R_θ^Δ corresponds to their real contrast.

5 Conclusions

In this paper we analyze the behavior of ultimate opening on blurred objects and see that it underestimates their contrast. We propose a solution, the Δ ultimate opening, that integrates series of residues, getting a much better contrast estimation. Moreover, Δ *UO* can be used as a detector of blurred contours, without any additional parameter. They are defined as the set of pixels that benefit from the integration.

We propose an efficient implementation of Δ *UAO*, as a simple modification from the max-tree based *UAO* implementation.

Interesting segmentation results for text segmentation in generic images are reported.

In the future, other criteria to combine residues can be studied. For example, the region stability, establishing some links with Maximal Stable Extremal Regions [11]. We also intend to analyze the influence of the chosen attribute on *UAO* result.

References

1. Beucher, S.: Numerical residues. *Image Vision Computing* 25(4), 405–415 (2007)
2. Retornaz, T., Marcotegui, B.: Scene text localization based on the ultimate opening. In: *International Symposium on Mathematical Morphology*, vol. 1, pp. 177–188 (2007)
3. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding* 64(3), 377–389 (1996)
4. Fabrizio, J., Marcotegui, B.: Fast implementation of the ultimate opening. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) *ISMM 2009*. LNCS, vol. 5720, pp. 272–281. Springer, Heidelberg (2009)

5. Leite, N.J., Guimarães, S.J.F.: Morphological residues and a general framework for image filtering and segmentation. *EURASIP J. Appl. Signal Process.* 2001, 219–229 (2001)
6. Outal, S., Beucher, S.: Controlling the ultimate openings residues for a robust delineation of fragmented rocks. In: *The 10th European Congress of Stereology and Image Analysis* (2009)
7. Hernández, J., Marcotegui, B.: Ultimate attribute opening segmentation with shape information. In: Wilkinson, M.H.F., Roerdink, J.B.T.M. (eds.) *ISMM 2009. LNCS*, vol. 5720, pp. 205–214. Springer, Heidelberg (2009)
8. Serra, J.: A lattice approach to image segmentation. *J. Math. Imaging Vis.* 24, 83–130 (2006)
9. Ronse, C.: Partial partitions, partial connections and connective segmentation. *J. Math. Imaging Vis.* 32, 97–125 (2008)
10. ICDAR, Robust reading and locating database (2003), <http://algoval.essex.ac.uk/icdar/TextLocating.html>
11. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *Proc. of the British Machine Vision Conference*, pp. 384–393 (2002)
12. Hernández, J.: Analyse morphologique d’images pour la modélisation d’environnements urbains, Ph.D. thesis, Mines ParisTech - C.M.M., Fontainebleau - France (December 2009)

A ΔUAO Implementation on Max-Tree

ΔUAO can be implemented easily and efficiently on a max-tree representation. As explained in [4], UAO can be computed in a single tree traversal, from the root to the leaves, with the following formula:

$$r_\lambda(\textit{node}) = \begin{cases} t(\textit{node}) - t(\textit{parent}) + r_\lambda(\textit{parent}) & \kappa(\textit{parent}) == \kappa(\textit{node}) \\ t(\textit{node}) - t(\textit{parent}) & \textit{otherwise} \end{cases} \quad (5)$$

where \textit{node} and \textit{parent} are two linked nodes of the max-tree, $t(\textit{node})$ the gray level associated to \textit{node} and $\kappa(\textit{node})$ its corresponding attribute.

In order to implement ΔUAO , the condition $(\kappa(\textit{parent}) == \kappa(\textit{node}))$ is replaced by $(\kappa(\textit{parent}) - \kappa(\textit{node})) \leq \Delta$ in the previous equation. The reason for that is the following: an opening of size $\kappa(\textit{node})+1$ will filter out the region corresponding to the \textit{node} , assigning to it the value of its \textit{parent} . Thus, the following opening producing a residue in this region would be the opening of size $\kappa(\textit{parent}) + 1$, that removes the \textit{parent} node. Therefore, openings from size $\kappa(\textit{node})+2$ to size $\kappa(\textit{parent})$ (both included) will not modify the region, leading to a series of $\kappa(\textit{parent}) - \kappa(\textit{node}) - 1$ null-residues. Given that, according to ΔUAO definition, residues should be integrated until finding a series of Δ null residues, the condition to this integration is set at $(\kappa(\textit{parent}) - \kappa(\textit{node}) \leq \Delta)$. The formula for r_λ^Δ computation becomes:

$$r_{\lambda}^{\Delta}(\text{node}) = \begin{cases} t(\text{node}) - t(\text{parent}) + r_{\lambda}^{\Delta}(\text{parent}) & \kappa(\text{parent}) - \kappa(\text{node}) \leq \Delta \\ t(\text{node}) - t(\text{parent}) & \text{otherwise} \end{cases} \quad (6)$$

The pseudo-code for ΔUAO is shown in algorithm 1. R_{θ}^{Δ} is computed from r_{λ}^{Δ} as explained in [4,12]. The process starts at the root node: $R_{\theta}^{\Delta}(\text{root})$ and $q_{\theta}^{\Delta}(\text{root})$ are initialized to zero. Then, function `ComputeNode` is called for each root child. `ComputeNode` function (see algorithm 2) computes $r_{\lambda}^{\Delta}(\text{node})$ and compares it with $R_{\theta}^{\Delta}(\text{parent})$. $R_{\theta}^{\Delta}(\text{node})$ keeps the maximum value between them: $R_{\theta}^{\Delta}(\text{node}) = \max(r_{\lambda}^{\Delta}(\text{node}), R_{\theta}^{\Delta}(\text{parent}))$. q_{θ}^{Δ} computation requires a particular attention. By definition it should be set at the size of the opening producing the highest residue. But several openings contribute to R_{θ}^{Δ} . Which size should be chosen for q_{θ}^{Δ} ? If $R_{\theta}^{\Delta}(\text{parent})$ is higher than $r_{\lambda}^{\Delta}(\text{node})$, $R_{\theta}^{\Delta}(\text{parent})$ and $q_{\theta}^{\Delta}(\text{parent})$ are propagated to the *node* (lines 14 and 15 in algorithm 2). Otherwise, if $r_{\lambda}^{\Delta}(\text{node})$ becomes $R_{\theta}^{\Delta}(\text{node})$, $q_{\theta}^{\Delta}(\text{node})$ is set at:

- $\kappa(\text{node}) + 1$, if the *node* does not belong to an integration series (*is Δ* is false) or if *node* is the first node of the integration series when a maximal residue is produced (*is Δ* is true but *q-propagation* is false; lines 20 and 21).
- $q_{\theta}^{\Delta}(\text{parent})$, until the end of the integration process (*is Δ* and *q-propagation* are both true; lines 18 and 19).

Thus, q_{θ}^{Δ} is set at the largest opening size involved in the integration process, when a maximum residue is produced. This size corresponds to the actual size of the detected region.

An example of r_{λ}^{Δ} computation, with $\Delta = 1$, for a synthetic profile of Fig. 7a is illustrated in Fig. 8. For the sake of simplicity, tree nodes are labelled with their gray level value (which is obviously not the case in the software implementation).

- The residue of the root node (0) is initialized to zero: $r_{\lambda}^{\Delta}(0) = 0$. $R_{\theta}^{\Delta}(0)$ and $q_{\theta}^{\Delta}(0)$ are also initialized to 0.
- Then, node 3 is processed: $r_{\lambda}^{\Delta}(3) = r_{\lambda}(3) = 3$. No integration is performed, since $\kappa(3) - \kappa(0) = 10 - 8 = 2 > \Delta$. $r_{\lambda}^{\Delta}(3) > R_{\theta}^{\Delta}(0)$, then $R_{\theta}^{\Delta}(3) = r_{\lambda}^{\Delta}(3)$ and $q_{\theta}^{\Delta}(3) = \kappa(3) + 1 = 9$.
- After that, node 5 is processed: $r_{\lambda}^{\Delta}(5) = r_{\lambda}(5) = 2$. No integration is performed, because $\kappa(5) - \kappa(3) = 8 - 6 = 2 > \Delta$. Since $r_{\lambda}^{\Delta}(5) < R_{\theta}^{\Delta}(3)$, R_{θ}^{Δ} and q_{θ}^{Δ} are propagated from the parent (node 3) to the child (node 5): $R_{\theta}^{\Delta}(5) = R_{\theta}^{\Delta}(3) = 3$ and $q_{\theta}^{\Delta}(5) = q_{\theta}^{\Delta}(3) = 9$.
- When computing $r_{\lambda}^{\Delta}(7)$, $r_{\lambda}(7)$ should be added to $r_{\lambda}^{\Delta}(5)$: $r_{\lambda}^{\Delta}(7) = r_{\lambda}(7) + r_{\lambda}^{\Delta}(5) = 2 + 2$, because $\kappa(7) - \kappa(5) = 6 - 5 = 1 \leq \Delta$. This residue is larger than $R_{\theta}^{\Delta}(\text{parent})$, then $R_{\theta}^{\Delta}(7) = r_{\lambda}^{\Delta}(7)$ and $q_{\theta}^{\Delta}(7) = \kappa(7) + 1 = 6$. The variable *q-propagation* is activated for the rest of the integration series.
- This integration continues till node 9 (attributes have consecutive values: from 6 to 3), leading to $r_{\lambda}^{\Delta}(9) = 2 + 2 + 1 + 1 = 6$. Thus $R_{\theta}^{\Delta}(8) = r_{\lambda}^{\Delta}(8) = 5$ and $R_{\theta}^{\Delta}(9) = r_{\lambda}^{\Delta}(9) = 6$.

Algorithm 1. Compute Delta Ultimate Attribute Opening

```

1 Compute $\Delta UAO()$ 
  // Initialization
2  $R_{\theta}^{\Delta}(root) = 0$ 
3  $q_{\theta}^{\Delta}(root) = 0$ 
4  $q\_propagation = 0$ 
  // Compute children
5 foreach  $child$  in  $Children(root)$  do
6    $ComputeNode(child, root, q\_propagation)$ 

```

Algorithm 2. Compute Node

```

7  $ComputeNode(node, parent, q\_propagation)$ 
  // Verify  $\Delta$  attribute
8  $is\Delta = (\kappa(parent) - \kappa(node)) \leq \Delta$ 
  // Compute residue  $r_{\lambda}^{\Delta}(node)$ 
9 if ( $is\Delta$ ) then
10    $r_{\lambda}^{\Delta}(node) = t(node) - t(parent) + r_{\lambda}^{\Delta}(parent)$ 
11 else
12    $r_{\lambda}(node)^{\Delta} = t(node) - t(parent)$ 
  // Compute  $R_{\theta}^{\Delta}(node)$  and  $q_{\theta}^{\Delta}(node)$ 
13 if ( $R_{\theta}^{\Delta}(parent) > r_{\lambda}^{\Delta}(node)$ ) then
14    $R_{\theta}^{\Delta}(node) = R_{\theta}^{\Delta}(parent)$ 
15    $q_{\theta}^{\Delta}(node) = q_{\theta}^{\Delta}(parent)$ 
16 else
17    $R_{\theta}^{\Delta}(node) = r_{\lambda}^{\Delta}(node)$ 
18   if ( $is\Delta \wedge q\_propagation$ ) then
19      $q_{\theta}^{\Delta}(node) = q_{\theta}^{\Delta}(parent)$ 
20   else
21      $q_{\theta}^{\Delta}(node) = \kappa(node) + 1$ 
  // Compute children
22 foreach  $child$  in  $Children(node)$  do
23    $ComputeNode(child, node, is\Delta)$ 

```

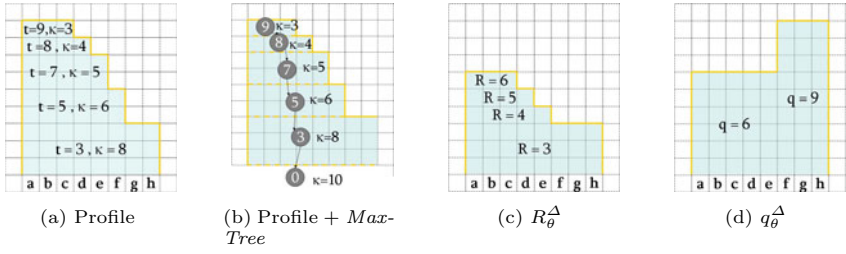


Fig. 7. (a) Profile with graylevels (t) and attribute (κ), (b) Max-Tree. (c)-(d) ΔUAO outputs

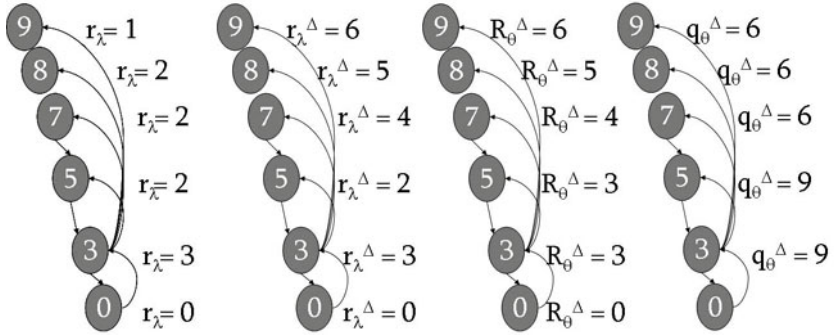


Fig. 8. ΔUAO on Max-tree