AUTO-DUAL CONNECTED OPERATORS BASED ON ITERATIVE MERGING ALGORITHMS

PHILIPPE SALEMBIER, LUIS GARRIDO and DAVID GARCIA
Universitat Politècnica de Catalunya, Barcelona, Spain *
Tel: (343) 401 7404, Fax: (343) 401 6447 Email: philippe@gps.tsc.upc.es

Abstract. This paper proposes a new set of connected operators that are auto-dual. Classical connected operators are analyzed within the framework of merging algorithms. The discussion highlights three basic notions: merging order, merging criterion and region model. As a result, a general merging algorithm is proposed. It can be used to create new connected operators and, in particular, auto-dual operators. Implementation issues are also discussed.

1. Introduction

Connected operators [5, 2, 4] are becoming popular because they simplify images without corrupting the shape information. It has been shown [8] that they interact with the image by merging the zones of the space where the signal is constant (called *flat zones*). As a result, they do not introduce new contours. These filters are extensively used in image analysis and segmentation applications [10, 3, 6, 1].

A large number of segmentation approaches, such as the *Split&Merge*, the *region growing* or the *watershed*, also rely on merging strategies. They have been created in a different context from that of connected operators and for different applications. Connected operators have been designed as a filtering tool whereas the others are devoted to segmentation. However, they all rely on the same fundamental merging process. The objective of this paper is to investigate the basic features of the merging process and to see how the connected operators can benefit from the "segmentation viewpoint". As major output of this study, we will create auto-dual connected operators. Moreover, the operators implementation will be discussed.

The organization of this paper is as follows. Section 2 reviews classical *connected operators* and a general merging strategy is proposed in section 3. An efficient implementation is discussed in section 4. Based on the general merging process, section 5 defines new auto-dual connected operators.

2. Classical connected operators

The most well known connected operators are the *opening by reconstruction* and the *area opening*. They are anti-extensive operators. In practice, this means that they only deal with bright components. Let us describe the filtering process on the Region Adjacency Graph (RAG) of the flat zones partition (see Fig. 1 where the numbers indicate the gray level values of the flat zones).

* This work has been supported by France Telecom under the contract 96ME22.

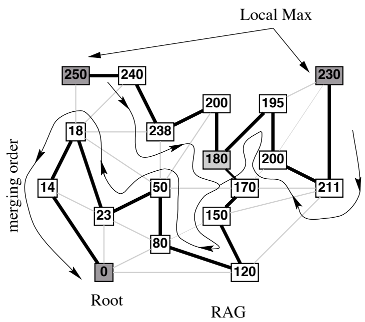

Fig. 1: RAG of flat zones

First, the regional maxima are computed. Then, the operator assesses a given criterion for each connected component of the regional maxima (for an area opening [9], the criterion is the number of pixels). If the criterion value is higher than a given threshold, the connected component is preserved, otherwise it is merged with the flat zone(s) of <u>closest lower</u> gray level value. This procedure is repeated on the new connected component resulting from the merging until the criterion value becomes higher than the threshold. Note that the merging order is defined by the absolute gray level value of the flat zones. The first pair of flat zones that is studied is the pair involving the highest gray level value. Then, progressively, flat zones of lower gray level values are processed.

Let us now discuss a region growing process. In this case, an initial set of regions is defined. Then, based on a similarity measure such as the mean gray level difference, neighboring pixels are progressively merged to create regions. Note that the merging order is not known at the beginning of the process. Indeed, after each merging, the algorithm has to look for the neighboring pixel of a region that minimizes its distance to the region. However, when one pixel is merged with a region, the distance between this region and the remaining neighboring pixels may change. As a result, the merging order can only be defined step by step during the merging process itself. This merging process is iterated until a termination criterion is reached.

Let us note that the algorithm used for connected operators has strong similarities with a region growing process. It also involves an initialization step defining a set of initial regions (regional maxima) and it also involves merging steps. There are, however, some important differences:

1. The merging order is known a priori in the case of connected operators (gray level value) and not in the case of region growing (on-line distance computation).
2. The second difference is that connected operators make use of a criterion to decide whether a given merging has to be done or not (the size of the connected components in our previous example). In fact, the merging order defines the order in which the pairs of flat zones are analyzed. However, each pair of flat zones can be merged or not. The region growing algorithm decides to merge all pairs of regions that are proposed by the merging order (until a termination criterion is reached). In contrast, the connected operator algorithm does not merge all pairs of flat zones but select some of them. This difference is rather natural, since a filter is a sieving tool, that is, it selects regions or flat zones with a given and specific characteristic.
3. Finally, in the case of classical connected operators, when two flat zones are merged, the lowest gray level value is assigned to resulting connected component. This assignation strategy means that there is an implicit modeling: each flat zone is represented by the lowest gray level value of its pixels in the original image. In the case of a region growing algorithm, the mean is generally used as region model.

3. General merging algorithm

Taking into account the interesting features of region growing and connected operators, let us define a general merging strategy. The proposed algorithm works on the RAG of flat zones which is a set of nodes representing flat zones and a set of links connecting two neighboring nodes. A merging algorithm on this graph is simply a technique that removes some of the links and merges the corresponding nodes. In the sequel, we assume that the merging is done in an iterative way. To completely specify a merging algorithm one has to define three notions:

1. The **merging order**: it defines the order in which the links are studied to know whether or not they should disappear. This order $\mathcal{O}(R_1, R_2)$ is a real value and is a function of each pair of neighboring flat zones R_1, R_2.
2. The **merging criterion**: each time a link is analyzed, the merging criterion decides if the merging has actually to be done or not. It is also a function $\mathcal{C}(R_1, R_2)$ of two neighboring flat zones R_1 and R_2, but it can only take two values ("merge" and "do not merge").
3. The **region model**: when two flat zones are merged, the model defines how to represent the union. Let us denote by $\mathcal{M}(R)$ this model.

In the case of opening by reconstruction or area opening, the merging order and the model are not auto-dual, that is, they do not commute with a modification of the gray level values of the type: $f \to -f$. In section 5, new auto-dual connected operators will be created by using this algorithm and by selecting an auto-dual merging order and an auto-dual model.

4. Efficient implementation

4.1. Merging algorithm on a RAG structure

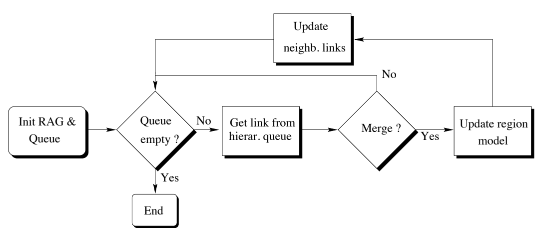

Fig. 2: Merging Algorithm

This section is devoted to the implementation of the merging algorithm described in section 3. The general scheme presented in Fig. 2 can be divided into two stages.

- "Initialization" of the RAG structure and of the hierarchical queue (see section 4.2). The first step consists in calculating the merging order for each pair of neighboring flat zones (called *link* in the following). Then, each link is inserted in the hierarchical queue with a priority defined by the merging order.

- "Merging" process: the algorithm extracts the link of highest priority from the queue. The next step decides whether the pair of flat zones corresponding to this link have to be merged or not. This decision relies on the "merging criterion". If the merging criterion decides not to merge the flat zones, the algorithm returns to the first step of the algorithm[1]. If the merging criterion decides to merge, the next step computes the region model of the union. To avoid redundant computations, a recursive algorithm is needed. "Recursive" here means that the model of the union should be computed from the models of the two initial flat zones. Once the flat zones have been merged, the values of the merging order of the neighboring links are updated. This implies the extraction of the corresponding links from the queue, the computation of the new priority and the insertion of the links in the queue with their new priority. At this point the merged RAG has been computed and updated: the iterative process starts again by checking if the queue is empty.

The key element of the implementation is a queue. In our context, its main features should be: first, fast access, insertion and deletion of an element and, second, no constraints on the dynamic range of the priority (floating point ordering).

4.2. Hierarchical queues and binary trees

Queues and hierarchical queues have been extensively used for fast implementation of morphological operators [10, 4]. For our merging algorithm, the main issue is the merging order that has to be constantly updated. In fact, the hierarchical queue should be updated and re-organized on-line. It can be seen as a dynamic hierarchical queue by contrast to the more classical "static" hierarchical queues. Another drawback of the classical hierarchical queue is that it is generally limited in the range of priority it can deal with and does not allow a floating point priority.

The solution proposed here is based on a binary tree. The basic idea of the hierarchical queues implementation with binary trees is depicted in Fig. 3. Each tree node represents a given priority. Furthermore, the sub-tree hanging on the left (right) side of each node involves nodes of lower (higher) priority. At each node, the links having the same priority are stored in a FIFO structure.

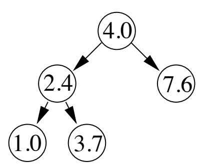

Fig. 3: Binary tree as a queue

To extract the links with highest priority, one begins with the root node and walks down the tree using the right branches until a node with no right branch is found. Search, insertion and deletion of nodes in the queue can be done in $O(log_2N)$ steps where N is the number of nodes in the tree [11].

One of the drawbacks of using binary trees to implement hierarchical queues is that the tree may degenerate. This problem happens when the merging order is not random. If, for example, the merging order is increasing, the resulting tree "leans" too much to the left and it turns into a simple FIFO queue. In this case, the $O(log_2N)$ efficiency does not hold anymore. Fortunately, there is a solution to this problem called "balanced trees". Efficient implementations of insertion and deletion preserving the tree balance can be found in [11]. The efficiency of the algorithm turns out to be very high. In practice, the computation time is about the same as the one for an anti-extensive connected operator or a watershed algorithm. For instance, the CPU time is about 1 second for a QCIF image on a 200MHz Pentium.

[1] Note that this decision is final in the sense that the corresponding pair of flat zones will never be merged.

5. Auto-dual connected operators

The objective of this section is to present new connected operators. As discussed in the introduction, one of our motivations is to create auto-dual operators. With the structure defined in sections 3 and 4, it is rather easy to create new connected operators that are auto-dual. To this end, one has to define the merging order $\mathcal{O}(R_1, R_2)$, the merging criterion $\mathcal{C}(R_1, R_2)$, and the region model $\mathcal{M}_R(x)$ in such a way that the resulting operator be auto-dual.

5.1. Region model: $\mathcal{M}_R$

The first choice that has to be made is how each flat zones is going to be modeled. For instance, in the case of an opening (a closing), each flat zone is modeled by the lowest (highest) gray level value of its pixels in the original image. Obviously, this model does not allow the creation of auto-dual operators. Two simple auto-dual models are the mean and the median. In order to allow a fast implementation of the merging process, the model of region $R = R_1 \cup R_2$ should be computed recursively from the models of the two merged regions R_1 and R_2. In the case of the mean model, one has to compute the weighted average of the mean of each flat zone:

$$\mathcal{M}_R = (N_1 \mathcal{M}_{R_1} + N_2 \mathcal{M}_{R_2})/(N_1 + N_2) \tag{1}$$

where N_1 and N_2 denote the number of pixels of each region. In the case of the median, one has simply to select the model of the largest flat zone [2]:

$$\begin{aligned} &\text{if } N_1 \; < \; N_2 \Longrightarrow \mathcal{M}_R = \mathcal{M}_{R_2}, \text{ if } N_1 > N_2 \Longrightarrow \mathcal{M}_R = \mathcal{M}_{R_1}, \text{ and} \\ &\text{if } N_1 \; = \; N_2 \Longrightarrow \mathcal{M}_R = (\mathcal{M}_{R_1} + \mathcal{M}_{R_2})/2 \end{aligned} \tag{2}$$

The median is known to be a robust estimator and, from our practical experience, it is indeed more robust than the mean. In fact, it rapidly defines areas where the model is stable. These areas can be seen as the core of the final flat zones. Then, small flat zones are progressively merged to the core without modifying the model. This property does not hold in the case of the mean. In the sequel, we assume that the median is used.

5.2. Merging order: $\mathcal{O}(R_1, R_2)$

The merging order defines the notion of objects. It can be seen as a measure of the likelihood that two neighboring flat zones belong to the same object. A natural measure is the squared error between the original image and the model in the region of support defined by $R = R_1 \cup R_2$: $\mathcal{O}(R_1, R_2) = \sum_{x \in R_1 \cup R_2} (f(x) - \mathcal{M}_{R_1 \cup R_2})^2$, if $f(x)$ denotes the gray level value of the original image at position x. This criterion actually allows the extraction of meaningful objects that are homogeneous in gray level, however it does not define precisely the contours. This drawback is related to the size dependence of the criterion. Indeed, small flat zones have a tendency to merge together since the squared error contribution of the union of two small flat zones is small compared to the contribution resulting from the merging with a large

[2] Note that the resulting model is not exactly the median of the original pixels since the median is computed in an iterative way.

flat zone. An alternative solution is to use the difference between the models of the two neighboring flat zones: $(\mathcal{M}_{R_1} - \mathcal{M}_{R_2})^2$ or the Mean Square Error (MSE): $\sum_{x \in R_1 \cup R_2} (f(x) - \mathcal{M}_{R_1 \cup R_2})^2/(N_1 + N_2)$. These criteria are size independent and provide a very good definition of the contours, however, they do not define in a robust way the flat zones themselves. In practice, they produce a few large flat zones surrounded by a very large number of tiny flat zones. To obtain a compromise between the two previous orders, we propose the following merging order:

$$\mathcal{O}(R_1, R_2) = N_1(\mathcal{M}_{R_1} - \mathcal{M}_{R_1 \cup R_2})^2 + N_2(\mathcal{M}_{R_2} - \mathcal{M}_{R_1 \cup R_2})^2 \qquad (3)$$

In the case of the median model and if R_1 is smaller than R_2, the model after merging is $\mathcal{M}_{R_2}$ and the order reduces to $\mathcal{O}(R_1, R_2) = N_1(\mathcal{M}_{R_1} - \mathcal{M}_{R_2})^2$. It is the squared difference between the models weighted by the size of the smallest flat zone.

5.3. Merging criterion: $\mathcal{C}(R_1, R_2)$

The function $\mathcal{O}(R_1, R_2)$ defines the order in which the flat zones have to be processed. Now, the objective of the merging criterion is to select among the set of possible flat zones a few that fulfill a given criterion. In the following, we illustrate two examples: area and PSNR.

Area merging criterion: The objective of this criterion is to remove from the original image all flat zones that are smaller than a given threshold. To this end, the criterion simply states that two flat zones have to be merged if at least one of them is smaller than a given threshold (the size is defined by the number of pixels).

The effect of the operator can be seen in Fig. 4. When the area threshold is set to 10 pixels, the resulting image contains most of the original information except the very small regions. For example, the texture of the fish has been removed. The process is a connected operator and the contour information of the objects that have not been removed is well preserved. Note that the image is made of flat zones of at least 10 pixels. The simplification effect can also be seen in the number of flat zones. When the area threshold is increased to 100 pixels, the same kind of effect can be observed. Note that the simplification effect of this area operator is different from that of an area opening or closing. The operator proposed here not only deals with bright or dark areas but also with transition areas.

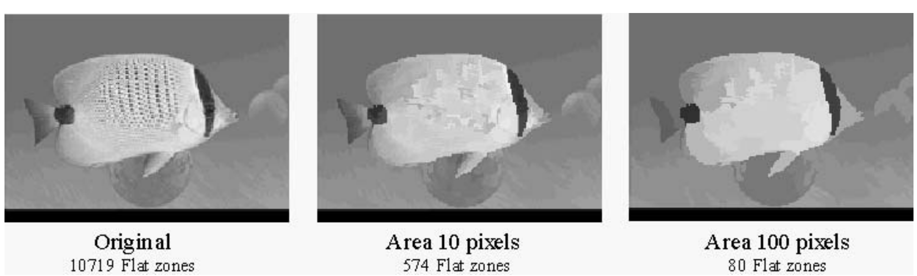

Fig. 4: Auto-dual area connected operator

PSNR merging criterion: for simplification or even segmentation, an interesting criterion is the Peak Signal to Noise Ratio (PSNR) between the original image and the modeled one. It is in fact a termination criterion which defines when to stop the merging process. The merging strategy defined in section 3 is particularly useful to compute pyramids of operators. Starting from the flat zone partition, several merging steps are performed. Note that with the implementation of section 4, it is possible to compute the full pyramid with only one run of the merging algorithm. Indeed, one has only to output some of the partitions created at intermediate stages of the merging process. This approach is therefore an efficient way to compute an entire hierarchy of partitions. An example of pyramid is shown in Fig. 5.

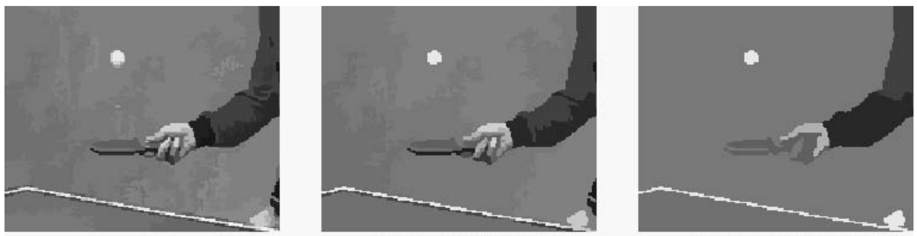

Fig. 5: Pyramid of PSNR connected operators

Let us discuss a potential use of the merging algorithm for *granulometric* analysis of images [7]. The approach consists in using a hierarchical filtering structure (classically, morphological opening) and in measuring an image characteristic (classically the integral of the image) at each filter output. This strategy allows the characterization of what has been removed by each filter. Consider now the PSNR connected operator and assume that PSNR values ranging from 40dB to 25dB are assigned to 16 levels of the hierarchy. Finally, suppose that we measure the number of flat zones of each processed image. The results are shown in Fig. 6. Three typical frames (frames #50, #200 and #270) of the *Foreman* sequence have been processed and the three corresponding curves are shown. These curves define how many flat zones are necessary to achieve a given PSNR. Intuitively, this number is function of the image "complexity": for simple frames with a few objects (homogeneous in gray level), the number of flat zones necessary to achieve a given PSNR is rather low. On the contrary, if the image involves a large number of contrasted objects, a high number of flat zones are necessary to reach the same PSNR. This intuitive discussion is confirmed by Fig. 6: among the three frames, the simplest one is the frame #200. It involves a few objects and the sky covers most of the frame. The corresponding curve is the lowest one. By contrast, the most complex frame is frame #270. It involves a fairly high number of contrasted objects and the corresponding curve is the highest one. Finally, frame #50 is of intermediate complexity and produces a curve in-between the two previous ones. Each curve (or a reduced set of curve points) can be used to characterize the image complexity.

6. Conclusions

This paper has focussed on the class of auto-dual connected operators based on merging techniques. The operator definition relies on three notions: first, the *merging order* that defines the notion of object homogeneity, second, the *merging criterion* that characterizes the set of flat zones we are interested in and, third, the *region model* that define how flat zones are represented.

The efficient implementation of a merging operator relies on a good management of the merging order. At each time instant, one should have, first, a very fast access to the next pair of flat zones to merge, and, second, an easy way to modify the various merging order values as the merging process goes on. The key element for a fast implementation is a dynamic hierarchical queue. A very efficient solution to this queue problem is to use a balanced binary tree.

Using this philosophy and implementation, new auto-dual connected operators based on area and on PSNR have been proposed and illustrated. These operators are particularly useful as simplification, pre-processing and segmentation tools.

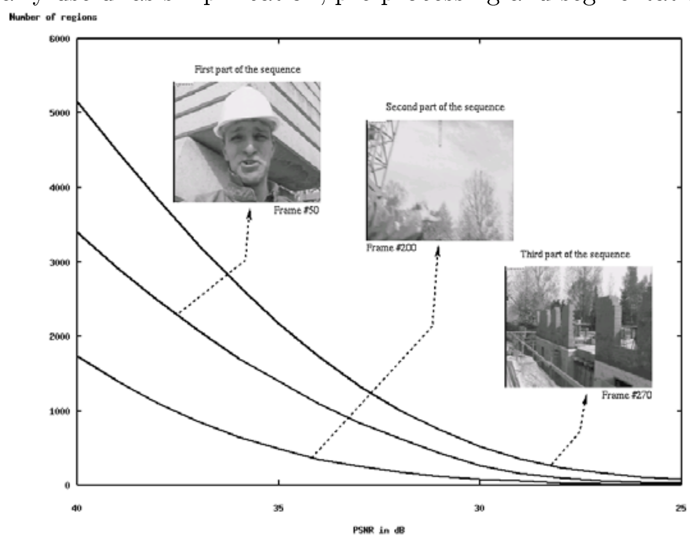

Fig. 6: Granulometric analysis with the PSNR connected operator

References

1. E. Breen and R. Jones. An attribute-based approach to mathematical morphology. In P. Maragos, R.W. Schafer, and M.A. Butt, editors, *International Symposium on Mathematical Morphology*, pages 41–48, Atlanta (GA), USA, May 1996. Kluwer Academic Publishers.
2. J. Crespo, J. Serra, and R.W. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 47(2):201–225, 1995.
3. B. Marcotegui, J. Crespo, and F. Meyer. Morphological segmentation using texture and coding cost. In I. Pitas, editor, *1995 IEEE workshop on Nonlinear Signal and Image Processing*, pages 246–249, Halkidiki, Greece, June 20-22 1995.
4. P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, April 1998.
5. P. Salembier and J. Serra. Flat zones filtering, connected operators and filters by reconstruction. *IEEE Transactions on Image Processing*, 3(8):1153–1160, August 1995.
6. P. Salembier, L. Torres, F. Meyer, and C. Gu. Region-based video coding using mathematical morphology. *Proceedings of IEEE (Invited paper)*, 83(6):843–857, June 1995.
7. J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1982.
8. J. Serra and P. Salembier. Connected operators and pyramids. In SPIE, editor, *Image Algebra and Mathematical Morphology*, volume 2030, pages 65–76, San Diego (CA), USA, July 1993.
9. L. Vincent. Grayscale area openings and closings, their efficient implementation and applications. In J. Serra and P. Salembier, editors, *First Workshop on Mathematical Morphology and its Applications to Signal Processing*, pages 22–27, Barcelona, Spain, May 1993. UPC.
10. L. Vincent. Morphological gray scale reconstruction in image analysis: Applications and efficients algorithms. *IEEE, Transactions on Image Processing*, 2(2):176–201, April 1993.
11. N. Wirth. *Algorithms & Data Structures*. Prentice-Hall, 1986.