MOTION CONNECTED OPERATORS FOR IMAGE SEQUENCES

Philippe Salembier, Albert Oliveras and Luis Garrido
Universitat Politècnica de Catalunya, Campus Nord - Módulo D5
C/ Gran Capità, 08034 Barcelona, Spain
Tel: (343) 401 74 04, Fax: (343) 401 64 47
E-mail: philippe@gps.tsc.upc.es

ABSTRACT

This paper deals with motion-oriented connected operators. These operators eliminate from an original sequence the components that do not undergo a specific motion (defined as a filtering parameter). As any connected operator, they achieve a simplification of the original image while preserving the contour information of the components that have not be removed. Motion-oriented filtering may have a large number of applications including sequence analysis with motion multi-resolution decomposition or motion estimation.

1 INTRODUCTION

Morphological filters by reconstruction, and more generally *connected operators*, are increasingly used in image processing [8, 5, 9, 1, 7, 6]. They are attractive in applications where the signal has to be simplified without loosing information about the contours. A large number of simplification criteria, such as size [8], area [11], dynamics [2], contrast, or complexity [4] can be obtained with these operators.

Motion information is a difficult issue in image sequence processing. Most of the time, motion is extracted from a local estimation that does not take into account the structure of the signal, that is the various objects in the scene. This is the case, in particular, for the popular block-matching or pel-recursive motion estimation algorithms [10]. The objective of this paper is to propose a filtering technique that leads to a different way of handling the motion information. The goal is to define a filtering tool allowing the simplification of the image following a motion criterion. In practice, the image components that do not undergo a specific motion should be removed.

In this paper, we propose to use a *connected operator* with a motion criterion to perform the simplification task. As will be seen, the operator offers a motion-oriented simplification effect while preserving the contour information of the non-simplified objects. This filtering technique can be used for a large set of applications including motion estimation, object tracking and motion-oriented multi-resolution decomposition.

The organization of this paper is as follows. The following section discusses the notion of *connected operators*. Section 3 is devoted to the definition of the motion criterion and the filtering process. One of the major theoretical issues of this operator is related to the non-increasingness of the criterion which may lead to instabilities in the filtered sequences. Finally, filtering examples and applications are reported in section 4.

2 CONNECTED OPERATORS

2.1 Binary connected operators

Let X denote a binary image. As defined in [9, 7], a binary *connected operator* ψ is an operator that only removes connected components of X or of its complement X^c. In the sequel, we restrict ourselves to the case of anti-extensive operators $(\forall X, \psi(X) \subseteq X)$. In this case, a binary *connected operator* is an operator that can only remove connected components of X.

The filtering process can easily be explained if a tree representation of the image is used. This approach is illustrated in Fig. 1. The original image X is composed of three connected

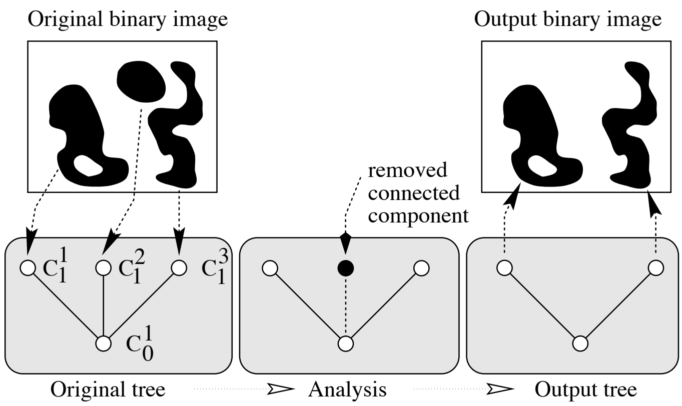


Figure 1: Binary connected operator

components. It can be represented by a tree structure with four nodes: the root node C_0^1 represents the set of pixels belonging to the background X^c, and $\{C_1^k\}_{1\leq k\leq 3}$ represent the three connected components of the image. In this representation, the filtering process consists in analyzing each node C_1^k by assessing the value of a particular criterion. Assume for example that the criterion consists in counting the number of pixels belonging to a node (area opening [11]). Then, for each node, the criterion value is compared to a given threshold λ and the node is removed if the criterion is lower than λ. In the example of Fig. 1, node C_1^2 is removed because its area is small and its pixels are moved to the background node C_0^1 (the connected component is removed). As can be seen, the tree links represent the pixels' migration (towards the father) when a node is removed.

Note that this process leads to a simplification of the image (some connected components are removed) as well as a perfect preservation of the contour information of the remaining components (components that are not removed are perfectly preserved). All anti-extensive binary connected operators can be described by this process, the only modification being the criterion that is assessed.

2.2 Gray-level connected operators

The extension of connected operators to gray-level images can be done via the notions of *flat zones* and the corresponding partition. The reader is referred to [9, 7, 6] for more theoretical details about this extension. Here, we present intuitively this extension by a simple generalization of the tree representation to the gray level case.

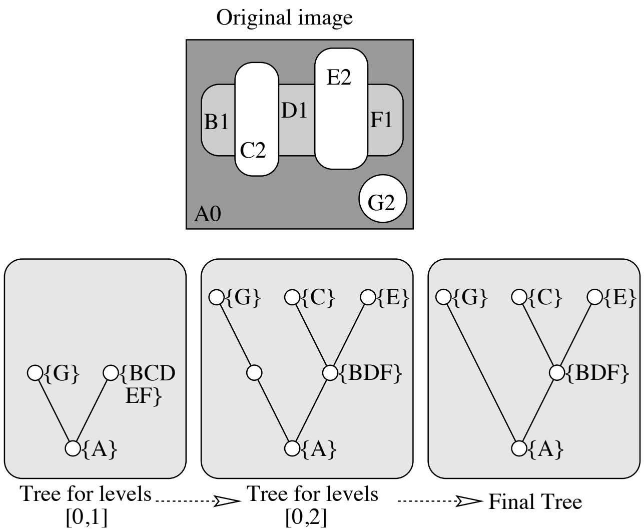


Figure 2: *Max-Tree* creation

The idea consists in creating recursively the tree by a study of thresholded versions of the image at all possible gray levels. An example is presented in Fig. 2. The original image is composed of seven *flat zones* (largest connected components where the signal is constant) identified by a letter $\{A, B, C, D, E, F, G\}$. The number following each letter defines the gray level value of the *flat zone*. In our example, the gray level values range from 0 to 2. In the first step, the threshold h is fixed to the gray level value 0. The image is binarized: all pixels at level $h = 0$, that is pixels of region A, are assigned to the root node of the tree $C_0^1 = \{A\}$. Furthermore, the pixels of gray level value strictly higher than h form two connected components: $C_1^1 = \{G\}$ and $C_1^2 = \{B, C, D, E, F\}$. This creates the first tree (for gray levels $[0, 1]$). Note that this procedure is the same as the one used for the binary image. In a second step, the threshold is increased by one $h = 1$. Each node $C_{h=1}^k$ is processed as the original image: consider, for instance, the node $C_1^2 = \{B, C, D, E, F\}$. All pixels belonging to this node that are at level $h = 1$ remain assigned to this node. However, pixels of gray level value strictly higher than h (here $\{E, C\}$) create two different connected components and are moved to two child nodes $C_2^2 = \{C\}$ and $C_2^3 = \{E\}$. The complete tree construction is done by iterating this process for all nodes k at level h and for all possible thresholds h (from 0 to the highest gray level value). The algorithm can be summarized saying that, at each node C_h^k, a "local" background is defined by keeping all pixels of gray level value equal to h and that the various connected components formed by the pixels of gray level value higher than h create the child nodes of the tree.

Note that in this procedure, some nodes may become empty. Therefore, at the end of the tree construction, the empty nodes are removed. The final tree is called a *Max-Tree* in the sense that it is a structured representation of the image which is oriented towards the maxima of the image (maxima are simply the leaves of the tree) and towards the implementation of anti-extensive operators.

The filtering itself is similar to the one used for the binary case. A criterion is assessed for each node $\mathcal{M}(C_h^k)$. Based on this value, the node is either preserved or removed. In this last case, the node's pixels are moved towards its father's node. At the end of the process, the output *Max-Tree* is transformed into a gray level image by assigning to the pixels of each node C_h^k the gray value h.

3 MOTION CONNECTED OPERATOR

The goal of this section is to present the motion criterion. As will be seen, this criterion is non-increasing. This issue is studied in section 3.2.

3.1 Motion criterion

Denote by $f_t(i, j)$ an image sequence where i and j represent the coordinates of the pixels and t the time instant. Our objective is to define a *connected operator* able to eliminate the image components that do not undergo a given motion. The first step is therefore to define the motion model giving for example the displacement field at each position $\{\Delta_i(i, j), \Delta_j(i, j)\}$. The field can be constant $\{\Delta_i, \Delta_j\}$ if one wants to extract all objects following a translation, but in general the displacement can depend on the spatial position (i, j) to deal with more complex motion models such as affine or quadratic.

The sequence processing is performed as follows: each frame is transformed into its corresponding *Max-Tree* representation and each node C_h^k is analyzed. To check whether or not the information contained in a given node is moving in accordance to the motion field $\{\Delta_i(i, j), \Delta_j(i, j)\}$ a simple solution consists in considering the region created by the pixels of the current node C_h^k and *all its children* and to compute the opposite of the Mean Displaced Frame Difference ($\mathcal{D}$) of this region with the previous frame. Note that, the opposite of the mean DFD is used so that the criterion value for a region that has to be preserved is higher than the corresponding value when the region has to be removed. More formally, if $\bar{C}_h^k$ denotes the current node C_h^k and all its children, the criterion can be expressed as:

$$\mathcal{D}_{f_t}^{f_{t-1}}(C_h^k) = \frac{-\sum_{i,j \in \bar{C}_h^k} |f_t(i,j) - f_{t-1}(i - \Delta_i, j - \Delta_j)|}{\sum_{i,j \in \bar{C}_h^k} 1} \qquad (1)$$

In practice, however, it is not very reliable to state on the motion of part of the image on the basis of only two frames. The criterion should have a reasonable memory of the past decisions. This idea can be easily introduced in the criterion by adding a recursive term. Two $\mathcal{D}$ are measured: one between the current frame f_t and the previous frame f_{t-1} and a second one between the current frame and the previous *filtered* frame

$\Psi(f_{t-1})$ (Ψ denotes the *connected operator*). The motion criterion is finally defined as:

$$\mathcal{M}(C_h^k) = \alpha \mathcal{D}_{f_t}^{f_{t-1}}(C_h^k) + (1-\alpha)\mathcal{D}_{f_t}^{\Psi(f_{t-1})}(C_h^k) \qquad (2)$$

where $0 \leq \alpha \leq 1$. If α is equal to 1, the criterion is memoryless, whereas low values of α allow the introduction of an important recursive component in the decision process. In a way similar to all recursive filtering schemes, the selection of a proper value for α depends on the application: if one wants to detect very rapidly any changes in motion, the criterion should be mainly memoryless ($\alpha \approx 1$), whereas if a more reliable decision involving the observation of a larger number of frames is necessary, then the system should rely heavily on the recursive part ($0 \leq \alpha \ll 1$). In the examples of section 4, the α value is set to 0.7.

3.2 Non-increasingness issue

The criterion defined by Eq. 2 is not increasing. Indeed, if a region X is included in a region Y, there is *a priori* no relations between the two $\mathcal{D}$. Note, for example, that the area criterion mentioned in section 2.1 is increasing because if $X \subseteq Y \Rightarrow Area(X) \leq Area(Y)$. Let us analyze the effect of having an increasing or a non-increasing criterion on the *Max-Tree* representation.

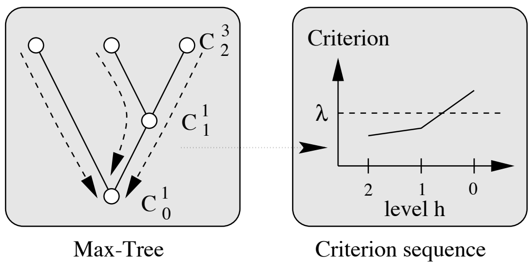


Figure 3: *Max-Tree* and *criterion sequence* for each local maximum

Consider a maximum of the image, that is a leaf node of the *Max-Tree*, and the sequence of all its ancestor nodes going down to the root node. In the example of Fig. 3, if we start by the maximum corresponding to C_2^3, the sequence is $C_2^3 \rightarrow C_1^1 \rightarrow C_0^1$. Consider now the sequence of the criterion values $\mathcal{M}(C_h^k)$ obtained by scanning successively all the ancestors of a maximum. In the example of Fig. 3, the *criterion sequence* starting from C_2^3 is $M(h) = [\mathcal{M}(C_2^3), \mathcal{M}(C_1^1), \mathcal{M}(C_0^1)]$ and is represented as a curve (function of h) on the right side. Note that the parameter h itself is decreasing because the nodes are scanned starting for the maximum and going down to the root. If the criterion is increasing, the *criterion sequence* is itself increasing and there is no problem to define the level h where the criterion is higher than a given limit λ. In this case, all nodes such that $\mathcal{M}(C_h^k) < \lambda$ are removed and the corresponding pixels are moved to the first ancestor node such that $\mathcal{M}(C_h^k) \geq \lambda$.

If the criterion is non-increasing, the *criterion sequence* $M(h)$ may fluctuate around the λ value and the definition of the set of nodes to remove is less straightforward. Two rules have been reported in the literature [3, 9, 7] to deal with the non-increasing case: the first one is called *by intersection or Min* and consists in preserving all nodes corresponding to levels h such that $\bigwedge_{0 \leq \nu \leq h} M(\nu) \geq \lambda$. The second rule is called *by union or Max* and consists in removing all nodes corresponding to levels h such that $\bigvee_{h < \nu < \infty} M(\nu) < \lambda$. The two rules are illustrated by Fig. 4.A. Experimentally, the *Min* rule is more robust and leads to more coherent decisions in time.

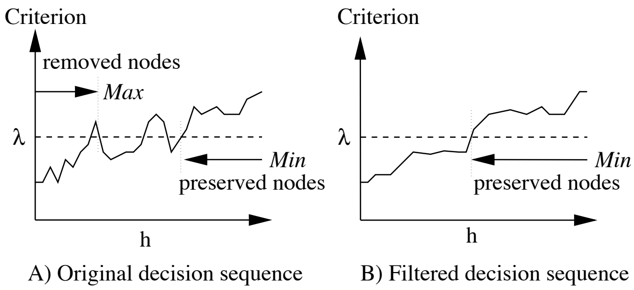


Figure 4: *Min* and *Max* rules on A) the original *decision sequence* and B) the filtered *decision sequence*

To filter the entire sequence, the decision and filtering processes are applied successively to all *Max-tree* representing each frame of the sequence. For sequence processing, the time coherence is primordial. A lack of coherence leads to random changes between elimination and preservation of some objects and is very annoying. To improve the robustness of the decision and of the filtering scheme, we propose to apply the *Min* rule on a filtered version of the *criterion sequence*. The observation of the motion *criterion sequences* reveals that the fluctuations around the decision value can be considered as an impulsive (decision) noise. We have therefore used a 1D median filter (with typically 3 or 5 samples) to remove the noise. This procedure is illustrated by Fig. 4.B (note that the Median filter reduces the impulsive noise but it does not necessarily remove it completely as shown in the example of Fig. 4).

4 EXAMPLES AND CONCLUSIONS

The first filtering example is shown in Fig. 5. The objective of the operator is to remove all moving objects. The motion model is defined by: $(\Delta_i, \Delta_j) = (0, 0)$. In this sequence, all objects are still except the ballerina behind the two speaker and the speaker on the left side who is speaking. The application of the connected operator $\Psi(f)$ described previously removes all bright moving objects (Fig. 5.B.). The application of the dual operator: $\Psi^*(f) = -\Psi(-f)$ removes all dark moving objects (Fig. 5.C.). The residue (that is the difference with the original image) presented in Fig. 5.D. shows what has been removed by the operator. As can be seen, the operator has very precisely extracted the ballerina and the (moving) details of the speaker's face.

The example illustrated in Fig. 6 shows a decomposition of the original image into three sequences: Objects with a translation of $(\Delta_i, \Delta_j) = (2, 0)$ (Fig. 6.B), still objects $(\Delta_i, \Delta_j) = (0, 0)$ (Fig. 6.C) and the remaining components (Fig. 6.D). This is a decomposition [5] of the original sequence in the sense that the sum of the three sequences restores the original sequence. As can be seen, the filtering has clearly separated the background and the two boats moving in two different directions.

The *motion connected operator* presented in this paper can potentially be used for a large set of applications. It opens the door in particular to different ways of handling motion information. Indeed, generally, motion information is measured

A) Original frame
f

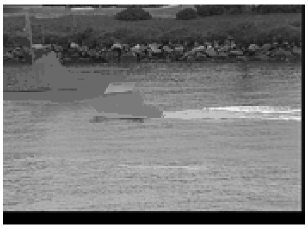
B) Motion connected
operator $\Psi(f)$

C) Dual operator
$\Psi^*(\Psi(f))$

D) Residue
f - $\Psi^*(\Psi(f))$

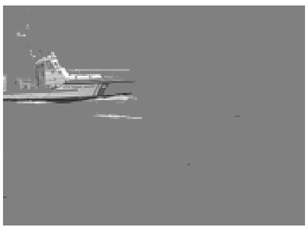
Figure 5: Example of motion connected operator preserving fixed objects

A) Original frame



B) Objects with
translation (2,0)

C) Objects with
translation (0,0)



D) Remaining objects

Figure 6: Example of motion-oriented decomposition (A = B + C + D)

without knowing anything about the image structure. *Connected operators* take a different viewpoint by making decisions on the basis of the analysis of all possible *flat zones*, that is of all possible structures of the image. By using *motion connected operators*, we can "inverse" the classical approach to motion and, for example, analyze simplified sequences where objects are following a known motion. The application of theses operators to motion-oriented segmentation of sequences as well as to motion estimation seems to be a very interesting field of research.

Acknowledgment

The authors are very grateful to Prof. J. Serra for the stimulating discussions on the topic of connected operators. This work has been partially supported by the CICYT TIC95-1022-C05-05 of the Spanish government.

References

[1] J. Crespo. *Morphological Connected Filters and Intra-region Smoothing for Image Segmentation.* PhD thesis, Georgia Institute of Technology, 1993.

[2] M. Grimaud. A new measure of contrast: the dynamics. In SPIE, editor, *Visual Communications and Image Processing'92*, volume 1769, pages 292–305, San Diego (CA), USA, July 1992.

[3] H. Heijmans. Theoretical aspects of gray level morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):568–592, 1991.

[4] A. Oliveras and P. Salembier. Generalized connected operators. In SPIE, editor, *Visual Communication and Image Processing, VCIP'96*, volume 2727, pages 761–773, Orlando (FL), USA, March 1996.

[5] P. Salembier and M. Kunt. Size-sensitive multiresolution decomposition of images with rank order based filters. *EURASIP Signal Processing*, 27(2):205–241, May 1992.

[6] P. Salembier and A. Oliveras. Practical extensions of connected operators. In *International Symposium on Mathematical Morphology, (Invited paper)*, Atlanta (GA), USA, May 1996.

[7] 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.

[8] J. Serra. *Image Analysis and Mathematical Morphology, Vol II: Theoretical advances.* Academic Press, 1988.

[9] 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.

[10] G. Tziritas and C. Labit. *Motion analysis for image sequence coding*, volume 4 of *Advances in Image Communication.* Elsevier, 1994.

[11] 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.