

Morphological Operators for Image and Video Compression

Philippe Salembier, Patrick Brigger, Josep R. Casas, and Montse Pardàs

Abstract—This paper deals with the use of some morphological tools for image and video coding. Mathematical morphology can be considered as a shape-oriented approach to signal processing, and some of its features make it very useful for compression. Rather than describing a coding algorithm, the purpose of this paper is to describe some morphological tools that have recently proved to be attractive for compression. Four sets of morphological transformations are presented: connected operators, the region-growing version of the watershed, the geodesic skeleton, and a morphological interpolation technique. Their implementation will be discussed, and we will show how they can be used for image and video segmentation, contour coding, and texture coding.

I. INTRODUCTION

IMAGE and video compression techniques generally rely on the results of the information theory. In this framework, compression is achieved by a decorrelation of the signal followed by quantization and entropy coding of the information to transmit. Decorrelation is obtained by using either predictive (DPCM, motion compensation) or transform (DCT, wavelets) techniques. For very high compression, there is an increasing interest in second-generation image compression techniques [14]. These techniques also eliminate the redundant information, but in addition try to take advantage of the properties of the human visual system. In particular, region-based compression methods describe the images or the sequences in terms of a set of regions, that is, a partition, and of some information for each region to be used by the receiver to reconstruct the image. This approach leads to contour/texture representations of the images. These techniques have been applied to the coding of still images [13], [14]. For sequences, region-based schemes have been developed in particular in [18], [24], [28], and [45].

In a region-based coding approach, the geometrical characteristics of the signal play an important role. For instance, the definition of the partition involves a segmentation that should ideally extract objects of the image or of the sequence. Obviously, objects are not only characterized by the correlation of their pixels but also by some geometrical properties.

Manuscript received December 1, 1994; revised November 13, 1995. This work was supported by the European Community through the MORPHECO project of the RACE program and the MAVI research network of the Human Capital and Mobility Program.

P. Salembier, J. R. Casas, and M. Pardàs are with the Department of Signal Theory and Communications of the Polytechnic University of Catalonia (UPC), Barcelona, Spain.

P. Brigger is with the Signal Processing Laboratory of the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

Publisher Item Identifier S 1057-7149(96)04181-4.

Another example is the coding of the partition. In this case, the information to transmit is purely geometrical. Finally, for very high compression of the texture information, only the most meaningful part of the signal can be transmitted. This meaningful part may be geometrically defined; let us mention as examples the minima and maxima of the signal, the lines of maximum and minimum curvature, etc. All these examples show that there may be a need for geometrical tools for image compression.

Classical linear signal processing tools are not well suited for a geometrical approach, and other tools coming from nonlinear signal processing or from computer vision may be attractive for this purpose. Mathematical morphology [36], [37] has been developed as a geometrical approach to signal processing, and our objective in this paper is to describe and discuss the usefulness of morphological tools in the context of image compression. The use of morphological tools for coding is becoming a very active field of research [35], [22], [31], [3], [27], [12], [7]. Rather than describing a complete coding/decoding scheme, this paper focuses on four morphological tools that have recently been defined and have proved to be useful for compression (the reader is referred to [31] and [35] for the description of complete coding algorithms). Moreover, these tools were selected because they cover the most important parts of a coding scheme. We will successively deal with the following morphological tools:

- *Connected operators* [39], [34]: This class of operators solves the problem of image simplification while preserving the contour information. The contour preservation property of connected operators is much better than that of linear, median, rank order, and classical morphological filters. They can be used for a large number of purposes in a coding scheme but they are specially useful for segmentation.
- *Region-growing version of the watershed* [21], [26], [32]: The watershed transformation is the classical morphological tool for segmentation [23]. However, it is generally applied on the gradient of the image. The use of the gradient results in a loss of information, which is not acceptable for coding, especially for moving sequences. The region-growing version of the watershed solves this problem. Moreover, it allows the introduction of complex criteria such as the contour complexity within the segmentation process.
- *Geodesic skeleton* [3]: Morphological skeletons have already been used for the coding of binary images [15], [12]. However, the coding of partitions is a different issue

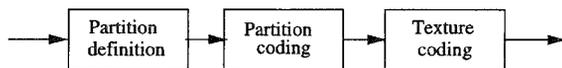


Fig. 1. Three steps of a coding process.

because the information to code cannot be decomposed into a set of objects and its complement, that is, the background. Since a contour belongs to at least two different regions, the description of each region by its skeleton results in a coding process where each contour is coded twice. The geodesic skeleton solves this problem and results in a more efficient coding.

- *Morphological interpolation* [40], [5], [33]: Geometrical interpolation techniques based on the notion of geodesic distance are very efficient tools to interpolate on non-regular grids. They can be used in combination with several image models to develop various texture coding strategies.

The organization of this paper is as follows. The next section introduces the various processing steps that are useful for coding applications, in particular segmentation, contour coding, and texture coding. Section III presents the notion of connected operators and their application to segmentation. Section IV is devoted to the region-growing version of the watershed and its use for segmentation in the framework of a coding application. Geodesic skeletons will be defined and applied to contour coding in Section V. Morphological interpolation will be presented in Section VI, and several texture-coding strategies will be proposed. Finally, Section VII concludes the paper.

II. STRUCTURE OF THE CODING PROCESS

This section describes some of the most important coding steps of a compression algorithm. The goal is not to propose a particular scheme but to discuss some structures and to define processing blocks where the morphological tools described in the sequel may be used.

On the upper level, one may consider that any coding algorithm involves three major steps represented in Fig. 1: *partition definition*, *partition coding* and *texture coding*. In the case of a block-based coding scheme, the *partition definition* corresponds simply to the division of the image into blocks and the *partition coding* can be removed because the receiver can restore this partition without any transmitted information. In the case of a region-based coding approach, the *partition definition* is the segmentation that extracts homogeneous regions. The homogeneity criterion can deal with information such as the gray-level values or the motion. The shapes of the resulting regions are arbitrary. They have to be transmitted to the receiver by the *partition coding*. In all cases, the texture is transmitted once the partition is known. Note that this approach is valid for both still image and sequence coding. In the case of sequence coding, partition and texture information is generally transmitted by using motion compensation. Let us describe more precisely some of the blocks involved in the three processing steps.

A. Iterative Segmentation

A fairly general segmentation structure is represented in Fig. 2 [27], [35]. It is an iterative segmentation structure that can be used either for intraframe or interframe segmentation. As can be seen in the upper part of Fig. 2, the segmentation is performed in several steps. Each step produces a new segmentation result $Partition(N)$ starting from a previous estimate of the segmentation $Partition(N-1)$.

In the case of intraframe segmentation, the *Original N* represents one of several versions of the frame to segment (for example, the original image at various levels of resolution). Each step of the iterative segmentation will improve the partition by introducing new regions and possibly by improving the contour position of known regions. As can be seen, the iterative segmentation corresponds to a hierarchical segmentation approach [31], [18]. For interframe segmentation, the *Original N* represents the frame at time N . In this case, the iterative segmentation structure will segment the frame N based on the knowledge of the partition at time $N-1$. This approach leads to a time-recursive segmentation where each step modifies the partition in order to follow the time evolution of past regions and, possibly, to introduce new regions [27]. Let us briefly describe the various steps used for the iterative segmentation (see Fig. 2).

- **Projection:** The *projection* block takes the previous segmentation result $Partition(N-1)$ and computes an estimate of the segmentation of the *Original N* without introducing new regions. In intraframe mode, if the resolution of the image has been modified, the projection will more precisely define the contour position. By contrast, if *Original N* does not depend on N , that is, if all the segmentation steps work on the full resolution image, the projection step can be reduced to the identity. In interframe mode, the projection defines the time evolution of the regions defined at time $N-1$. It is a temporal linking of the regions.
- **Coding and residue computation:** After projection, missing regions that are visually important should be introduced. To this goal, the effect of the coding process (partition and texture) should be estimated and the regions that are not well coded should be extracted. One solution consists in actually coding the image (*Coding* block) and in computing the difference between the coded image and the original (*Residue* block). The result is called the *residue* frame. It concentrates all the information about the regions that are not well coded with the current partition. To introduce new regions, the residue is segmented by the three following steps.
- **Simplification:** In this step, the residue is simplified to make it easier to segment. The simplification controls the *nature* and *amount* of information that is kept for segmentation, that is, the characteristics of the new regions to be introduced. Different simplification filters can be used depending on the segmentation criterion.
- **Feature extraction:** The goal of this step is to detect the presence of homogeneous regions, that is, to assess the local homogeneity. The feature extraction output

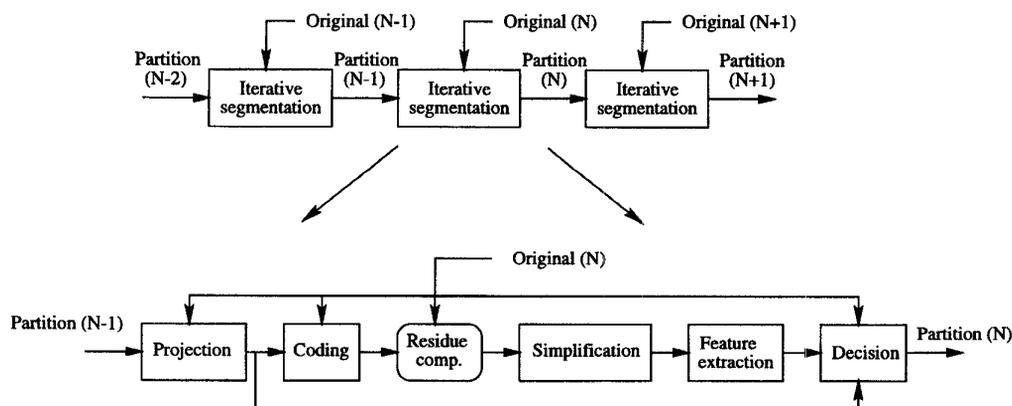


Fig. 2. Iterative segmentation process and description of a segmentation step.

can be a set of markers identifying the interior of the regions that will be segmented. In practice, a marker is a connected component of the image with a specific gray-level value indicating the number of the region. The marker defines the set of pixels that surely belong to the region and, in general, it defines the major part of the region interior. Note that the feature extraction technique depends on the segmentation criterion and, therefore, on the simplification filter.

- Decision:** After feature extraction, the number and the interior of the regions to be segmented are known. However, a large number of pixels are not assigned to any region. These pixels correspond to uncertainty areas mainly concentrated around the contours of the regions. Assigning these pixels to a given region can be viewed as a *decision* process that precisely defines the partition. This *decision* process only deals with the new regions, and it has to be constrained to take into account the partition that was defined by the *projection* step.

In Section III, we will show how mathematical morphology can be used in this segmentation process. We will see in particular that *connected operators* are extremely useful for *simplification* and *feature extraction* [31], [32] and that the *region-growing version of the watershed* is very attractive for both the *projection* and the *decision* [27], [26].

B. Contour Coding

Fig. 3 illustrates the transmission of the partition information by the classical motion predictive technique: Based on the previous partition image stored in the *contour memory* and on the motion information, a predicted partition is created by the *contour compensation* block. Its difference with the current partition defined by the segmentation is computed. Since we are dealing with partitions, the difference operator has to be considered as a set difference. The difference, called *contour error*, is simplified, coded, and transmitted to the receiver. Note that the coding of a still image or of an intraframe can be considered as a special case where no compensation is performed.

Morphological tools can be used at least at two different levels of this structure. First, they can be used as simplifica-

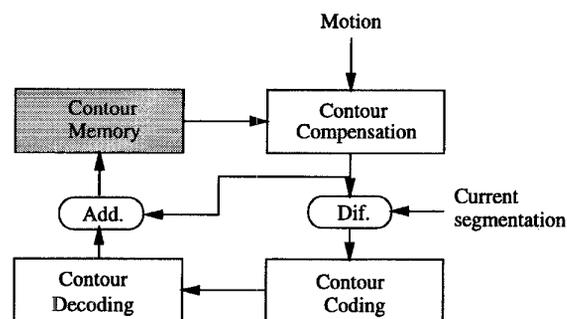


Fig. 3. Structure of the contour coding.

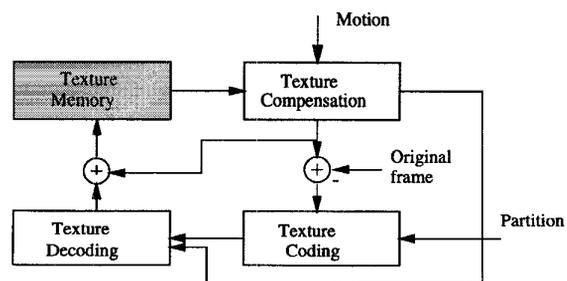


Fig. 4. Structure of the texture coding.

tion tools to remove small prediction errors that may result expensive in terms of coding but are not visually important. *Connected operators* are well suited for this purpose. Second, the partition errors that have to be transmitted after compensation can be represented either by their contours or by their shape. In the second case, *geodesic skeletons* are very efficient tools to code this kind of information [3].

C. Texture Coding

Once the partition has been restored (either transmitted or computed) in the receiver, the texture or color information has to be coded. A popular strategy follows the motion predictive technique illustrated in Fig. 4: Based on the previous texture image stored in the *texture memory* and on the motion information, a compensated texture image is created by the *texture compensation* block and its difference with the original

TABLE I
MORPHOLOGICAL TOOLS AND THEIR USE FOR CODING

	Connected operators	region growing watershed	Geodesic skeleton	Morphological interpolation
Projection		Yes		
Simplification	Yes			
Feature extraction	Yes			
Decision		Yes		
Contour coding	Yes		Yes	
Texture coding				Yes

texture frame is computed. The difference, called *texture error*, is coded and transmitted to the receiver. The final image is created by adding the compensated texture and the coded error. As before, this scheme is also valid for still images if no compensation is performed. Moreover, depending on the partition definition, the approach is adequate for block-based as well as region-based coding. In Section VI, the usefulness of *morphological interpolation* for texture coding will be presented and illustrated.

Table I summarizes the various blocks that have been described and indicates where the morphological tools described in the sequel can be used.

III. CONNECTED OPERATORS AND SEGMENTATION

A. Classical Linear and Nonlinear Tools for Image Simplification

In the context of segmentation, image simplification is generally used to eliminate the noise and to remove part of the signal that is of no interest for the segmentation process. The most classical simplification tool in signal processing is a linear lowpass filter. However, it is well known that this filter blurs edges and does not preserve the contour information. For a segmentation application, it is of course of prime importance to preserve the contour information. The problem of finding a simplification tool able to preserve the contour is a very active field of research. Many nonlinear filters such as median, rank-order, and morphological filters have been proposed. However, even if these filters give good results for 1-D signals, their performances deteriorate strongly for 2-D signals. Most of the time, the results are strongly influenced by the choice of the window of the filter. In the following section, we will show why connected operators can solve the problem of simplification and contour preservation [39], [34].

B. Connected Operators

The first connected operator reported in the literature is known as *opening by reconstruction*. It appeared experimentally for binary images in [11]. Initially, it consisted of eroding a binary image by a connected structuring element and in reconstructing all connected components that had not been totally removed by the erosion. It was called *opening* because it is an increasing, anti-extensive and idempotent process. It therefore possesses the three fundamental properties of an algebraic opening. Moreover, it was called *by reconstruction*

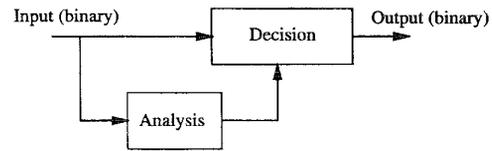


Fig. 5. Structure of a binary connected operator.

because it involves a reconstruction process of the connected components that have not been totally removed by the erosion. On this very simple example, one can see that the *binary opening by reconstruction* has the fundamental property of simplifying the signal while preserving the contour information. Indeed, the connected components of the binary image are either totally eliminated (the simplification effect) or perfectly preserved (the contour preservation).

The original idea of binary opening by reconstruction relies on a separation of an analysis process and of a decision process as illustrated in Fig 5. The simplification is basically a binary decision process stating which connected components have to be preserved and which have to be removed. In the original example of [11], the selection is done by the computation of an erosion; however, it can be extended to a large number of criteria such as the area, the Ferret diameter, etc.

In [39] and [34], the concept of binary connected operators is formally defined as follows: first, a connectivity has to be defined. In practice, the definition of the connectivity reduces to the definition of a local neighborhood system describing the connections between adjacent pixels. The classical choices involve four, six, or eight connectivity. Once the connectivity has been selected, the notion of connected operators can be defined as follows.

Binary Connected Operators: A binary operator ψ is said to be connected when for any binary image X , the symmetrical difference $X \setminus \psi(X)$ is exclusively composed of connected components of X or its complement X^c .

This is exactly the case of the binary opening by reconstruction, which acts only by preserving or removing connected components. The extension of the notion of binary connected operators to gray-level connected operators relies on the concept of partition [39], [34]. Note that the extension cannot be done directly because the connectivity has no equivalent in the case of gray-level functions. Let us recall that a partition of the space E is a set of connected components $\{A_i\}$, which are disjoint, and the union of which is the entire space. Each A_i is called a partition class. Moreover, a partition $\{A_i\}$ is said to be finer than another partition $\{B_j\}$ if any pair of points belonging to the same class A_i also belongs to a unique partition class B_j . Consider now a binary image and define its *associated partition* as the partition made of the connected components of the binary sets and of their complements. The definition of connected operators can be expressed with associated partitions.

Binary Connected Operators via Partition: A binary operator ψ is connected if and only if, for any binary image X , the associated partition of $\psi(X)$ is less fine than the associated partition of X .

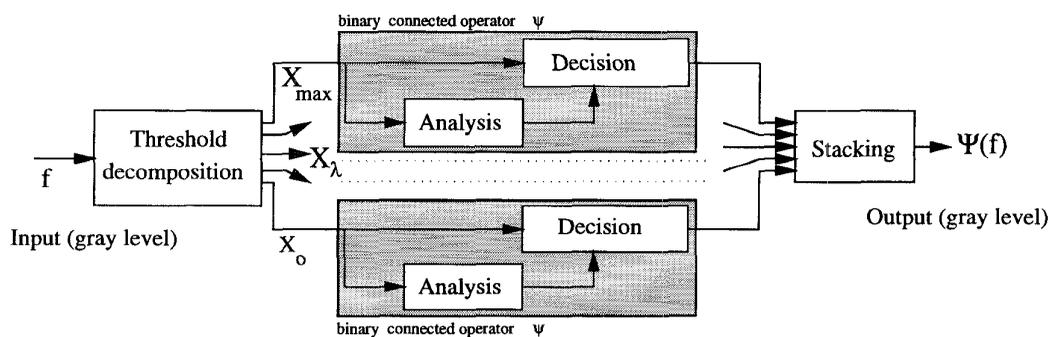


Fig. 6. Gray-level connected operator by threshold decomposition.

The concept of gray-level connected operators can be introduced if we define a partition associated to a function. To this end, the use of *flat zones* was proposed in [39] and [34]. The set of flat zones of a gray-level function f is the set of the largest connected components of the space where f is constant (note that a flat zone can be reduced to a single point). It can be demonstrated [39] that the set of flat zones of a function constitutes a partition of the space. This partition is called the *partition of flat zones* of a gray level function. It leads to the following formal definition.

Gray-Level Connected Operators: An operator Ψ acting on gray-level images is connected if, for any function f , the partition of flat zones of $\Psi(f)$ is less fine than the partition of flat zones of f .

There are several ways of creating gray-level connected operators. The simplest one consists of extending a binary connected operator. Indeed, as shown in [16], [37], and [10], any binary operator can generate a gray-level operator by *threshold decomposition* and *stacking*. This procedure is illustrated in Fig. 6. The threshold decomposition generates one binary image X_λ for each possible gray-level value λ , that is, 2^N binary images if the gray levels are quantized with N bits. Note that each binary image X_λ is associated to a specific gray level λ . Each binary image is processed by a binary connected operator ψ . Finally, the stacking consists in reconstructing a gray-level image $g = \Psi(f)$ from the set of binary images $\psi(X_\lambda)$ as follows:

$$g = \Psi(f) = \sup_{\lambda} \left(\bigcap_{\mu < \lambda} \psi(X_\mu) \right). \quad (1)$$

Note that if the binary connected operator ψ is increasing, the stacking can be simplified as follows:

$$g = \Psi(f) = \sup_{\lambda} (\psi(X_\lambda)). \quad (2)$$

Following this procedure, it can be shown [39], [34] that the resulting gray-level operator Ψ is a connected operator because the partition of flat zones of f is always finer than the partition of flat zones of $\Psi(f)$. The processing structure illustrated in Fig. 6 explains why gray-level connected

operators simplify while preserving the contour information. Indeed, as in the case of binary connected operators, a binary decision process states whether a flat zone has to be preserved or removed. Moreover, the decision process is separated from the reconstruction process.

C. Examples of Connected Operators

As examples of gray-level connected operators, let us describe the *gray-level opening by reconstruction*, the *area opening*, and the *h-max operator*.

- **Gray level opening by reconstruction:** As discussed previously, this filter consists of preserving all connected components (after threshold decomposition) that are not totally removed by a structuring element of size h . The gray-level output image is restored by stacking. This opening has a size-oriented simplification effect: it removes the bright image components that are smaller than the structuring element. By duality, a closing by reconstruction can be defined. Its simplification is similar to that of the opening but on dark components.
- **Gray-level area opening [43]:** This filter is similar to the previous one except that it preserves the connected components that have a number of pixels larger than a limit h . It is also an opening that has a size-oriented simplification effect, but the notion of size is different from the one used in the opening by reconstruction. By duality, an *area closing* can be defined.
- **h-max operator:** This operator differs from the previous ones only by the way it preserves the connected components after threshold decomposition. The criterion here is to preserve a connected component of the binary image X_λ if and only if this connected component hits one connected component of the binary image $X_\lambda \cap X_{\lambda+h}$. This is an example where the criterion involves two binary images obtained at two different threshold values. The simplification effect of this operator is contrast-oriented in the sense that it eliminates image components with a contrast lower than h . Note that the *h-max* is an operator and not a morphological filter because it is not idempotent. By duality, the *h-min operator* can be defined.

The definition of the previous connected operators has been done by using the scheme of Fig. 6. However, in practice, it

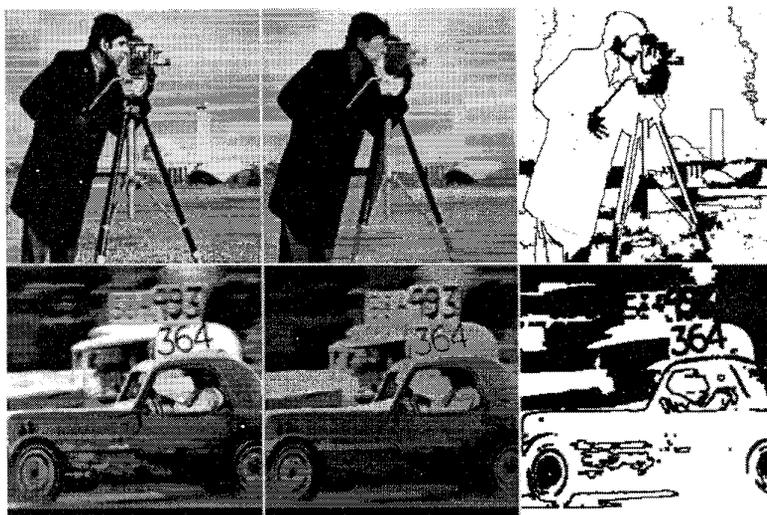


Fig. 7. Example of simplification and feature extraction with gray-level connected operators. First row: original image, simplification by an opening by reconstruction followed by a closing by reconstruction, size-oriented feature extraction. Second row: original, simplification by a h -max operator, positive contrast feature extraction.

is not necessary to use threshold decomposition and stacking to build a gray-level connected operator.

D. Implementation of the Reconstruction Process

The main bottleneck in the implementation of a gray-level connected operator is the reconstruction process. Indeed, the selection step is generally easier to compute. For instance, in the case of opening by reconstruction, the selection is performed by a simple gray-level erosion. In the case of the h -max operator, it relies on a simple subtraction of a constant value h from the original signal. The selection step produces what is generally called a *marker* image (because it indicates the connected components of the original signal that should be preserved).

Assume that the original and marker images are known and that we want to compute the reconstruction of the original image starting from the marker image. The most efficient reconstruction algorithm relies on the definition of a clever scanning of the image and are implemented by first-in-first-out (FIFO) queues. A review of the most popular reconstruction algorithms can be found in [44]. Here, let us describe a simple but efficient one.

The basic idea of the algorithm is to start from the regional maxima of the marker image and to propagate them *under* the original image (this reconstruction is known as a *positive* reconstruction and, by duality, a *negative* reconstruction can be defined). The algorithm works in two steps. The first one corresponds to the *initialization* of the queue and the second one performs the *propagation*.

- *Initialization* consists of putting in the queue the location of pixels that are on the boundary of the regional maxima of the marker image. Regional maxima are the set of connected components where the image has a constant gray-level value and such that every pixel in the neighborhood of the regional maxima has strictly a lower value.

Algorithms to compute regional maxima can be found in [42].

- *Propagation* extracts the first pixel x from the queue (note that x is a pixel of the marker image). Then, it assigns to each of its neighbors y , that have a strictly lower gray-level value than x , the minimum between the gray-level value of x and the gray-level value of the pixel of the original image at the same location as y . Finally, the pixel y is introduced in the queue. This propagation process has to be carried on until idempotence. This propagation process is very efficient because the image pixels are preprocessed only once.

E. Application of Connected Operators for Segmentation

Connected operators are very useful for segmentation, in particular, in the framework of coding. Indeed, for coding applications, the segmentation should be constrained and should extract only the most important regions of the images. *Opening (closing) by reconstruction* or h -max (h -min), respectively, eliminates small or low-contrasted regions. They allow a representation of the image by its regions of large size or of high contrast, which generally correspond to visually important image components. Moreover, they are suitable for simplification because they remove part of the information while preserving the contour information of the remaining image components. Both operators are illustrated by Fig. 7. The upper row shows a size-oriented simplification by an opening by reconstruction followed by a closing by reconstruction, whereas the lower row illustrates a contrast-oriented simplification by an h -max operator.

As discussed previously, these operators interact with the signal through the notion of flat zones. This property makes connected operators also very attractive for feature extraction. Indeed, in the case of size-oriented segmentation, the feature extraction has to detect large flat zones. This can be done by labeling all large flat zones after simplification by a connected

operator. In the case of contrast-oriented segmentation, a similar technique can be used. Consider a simplification with an h -max operator; the flat zones of high contrast can be identified by taking all flat zones after simplification where the difference between the original image and the simplified one is equal to h , at least in one point [30]. The images on the right side of Fig. 7 illustrate the feature extraction. In these examples, markers are represented in white whereas uncertainty areas are in black.

IV. WATERSHEDS AND SEGMENTATION

A. Classical Morphological Approach to Segmentation

The classical approach to segmentation [23] is to perform a feature extraction (generally called a *marker extraction*) and to use the watershed algorithm on the gradient of the image to segment. The watershed defines a *catchment basin* for each gradient minimum that has been identified by the marker extraction. This approach is not suitable for coding applications. Indeed, the use of the gradient results in a loss of information because if the original signal involves transitions, its morphological gradients are either biased (gradient by erosion $g^- = f - \epsilon^1(f)$ or by dilation $g^+ = \delta^1(f) - f$) or thick ($g = \delta^1(f) - \epsilon^1(f)$).¹ In the case of still images, this phenomenon is not extremely annoying. However, in the case of moving images, the use of the gradient results in a much larger loss of information [32]. Indeed, in the temporal direction, the thickness of the gradient depends on the motion of the objects and can become very large. Therefore, the use of the gradient should be avoided and the watershed algorithm has to be modified to work directly on the signal and not on its gradient.

B. The Region-Growing Version of the Watershed

The idea of using the watershed algorithm directly on the signal to segment was proposed in [21] to deal with color images and modified in [32] to improve the algorithm precision. The resulting watershed algorithm is a region-growing algorithm in the sense that it starts from the markers that identify the interior of the regions and extends them until they occupy all the available space.

Efficient implementations of watershed algorithms require a clever scanning of the images defined by *hierarchical queues*. A hierarchical queue is a set of FIFO queues with different priorities. The elements processed by the queue are pixel positions (the queue is used to define the scanning). This structure allows the representation of a double ordering: Pixels are put into one of the queues depending on a given priority. The first pixel to be pulled out of the queue is the first one that has entered the queue of highest priority. Then, successively, all pixels in the queue of highest priority are extracted. Finally, if the queue of highest priority is empty, the next pixel to be extracted is the first pixel of the first nonempty queue. Now, the region-growing version of the watershed can be simply implemented with these queues. The algorithm works

¹ δ^1 and ϵ^1 stand for the dilation and the erosion of size one, that is the smallest size on the digital grid.

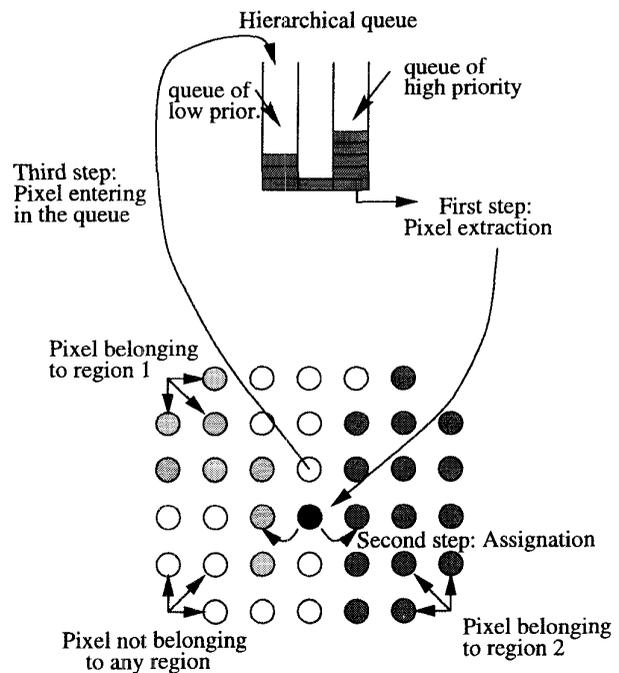


Fig. 8. Implementation of the watershed algorithm with a hierarchical queue.

in two distinct steps: queue *initialization* and *region-growing*, as follows.

- *Initialization* consists of putting the location of all pixels corresponding to the regions' interior in the queue of highest priority. The highest priority queue is used because the priority corresponds to the certainty with which a pixel belongs to a given region.
- *Region-growing* consists of extracting a pixel from the queue: If the pixel does not yet belong to a region, we know because of the filling procedure that it has at least one neighboring region. Therefore, a *distance* between the current pixel and each neighboring region is computed. The pixel is assigned to the region corresponding to the smallest distance. Then, if the current pixel has some neighbors that do not belong to any region, these neighbors are put in the queue with a priority defined as their distance to the region of the current pixel. As can be seen, any pixel that is put in the queue has at least one neighboring region. This is why it is possible to make a decision concerning this pixel when it will come out of the queue. The region-growing procedure is illustrated in Fig. 8 in the case of 2-D images. Note that this algorithm can be used for sequences viewed as 3-D signals. The only difference is the definition of the neighborhood system.

An important parameter of the algorithm is the distance function. Ideally, this distance estimates the certainty with which the pixel belongs to a region. However, for coding applications, it is useful to include in the distance a contour complexity criterion in order to limit the contour coding cost. We propose to define the distance as a weighted sum of the gray-level difference between the pixel gray-level value x and the mean of the region R and a term proportional to the length



Fig. 9. Example of decision with the region-growing version of the watershed. First row: original image, markers, decision with $\alpha = 1.0$. Second row: decision with $\alpha = 0.7$, decision with $\alpha = 0.4$, decision with $\alpha = 0.0$.

TABLE II
COST OF CONTOUR CODING AS A FUNCTION OF α

α	1.0	0.7	0.4	0.0
bits for contour	11816	10400	8200	4616

increment of the region contour $\Delta\partial R$. The increment of the region contour can be locally computed by considering the number of contour points that are added/removed each time a pixel is assigned to a specific region. The distance function is therefore defined by

$$d(x, R) = \alpha \left(x - \left(\frac{\sum_{y \in R} y}{\sum_{y \in R} 1} \right) \right) + (1 - \alpha) \Delta\partial R. \quad (3)$$

Note that if the distance (or priority) is defined as the gray level x (that is if $d(x, R) = x$), the algorithm is the classical watershed working on the gradient [23].

C. Application to Decision and Projection

Fig. 9 illustrates the use of the region-growing version of the watershed algorithm for the *decision* process of the segmentation. This figure shows an original image, a set of markers, and four segmentation results depending on the α parameter used to compute the distance. When α is equal to one, the distance only depends on the gray-level information of the images and the resulting contours can be complicated and therefore expensive to code. By contrast, if α is equal to zero, the segmentation does not take into account the gray levels of the image; it only extends the markers to minimize the contour complexity. Note that in this example, a size-oriented marker extraction was done. As a result, only large regions are segmented and some small details are not extracted (the mouth, part of the eyes, etc). To extract these details, a contrast-oriented segmentation can be used. In order to judge the

influence of the α parameter, the regions have been filled with the mean value of the original image and the corresponding partitions have been coded using the modified chain code technique proposed in [17]. Table II gives the number of bits necessary to code the partitions. As can be seen, the use of an α value of 0.7 allows a reduction of 10%–15% in the number of bits with hardly visible modifications of the partition.

The *projection* step described in Section II can also be computed by the watershed algorithm. The only modification is to consider the signals as 3-D signals. The process is illustrated by Fig. 10. Denote F_{t-1} and F_t the frames at time $t-1$ and t . Assume that the segmentation at time $t-1$, S_{t-1} , is known. The projection objective is to estimate the segmentation S_t at time t without introducing new regions. For this purpose, two 3-D signals are constructed. Frames F_{t-1} and F_t are grouped together to form a temporal block \mathbb{F} of size two in the time direction. Similarly, the frame S_{t-1} is grouped with an empty frame S_o representing an entire frame of uncertainty. The resulting 3-D signal denoted \mathbb{S} is considered as the set of markers that should be used to segment the signal \mathbb{F} . Each pixel of the uncertainty area (that is of frame S_o) is assigned to a region of frame S_{t-1} based on the distance criterion described previously. In 3-D, the contour complexity criterion controls the temporal stability of the contours.

Fig. 11 presents an example of a complete interframe segmentation. The first row shows three original images from the Foreman sequence. The second row shows the segmentation obtained using only the projection step. That is, the regions of the first image are extended into the new frames, but no new regions are introduced. This segmentation is coded here simply with the mean value of every region and the residue with the original frame is computed. From this residue, contrasted regions can be extracted. The result after this extraction of new regions, coded with a second order polynomial is shown in the third row. Only one new region corresponding to the mouth has appeared in the third frame. This region could not be produced from the previous image. Finally, in the fourth

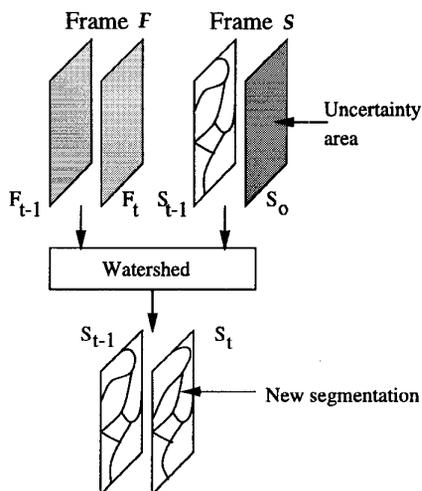


Fig. 10. Projection using the watershed algorithm.

row we have represented the final partition with different gray levels for each region. The new region can be identified, since it is represented by the brightest gray-level value.

V. GEODESIC SKELETON AND CONTOUR CODING

A. Contour-Oriented Coding of Contour Errors

Partition coding techniques can be classified into contour- or shape-oriented approaches. The first set of techniques represents the partition by describing its contours, whereas the second one deals with the shape of the regions.

The most popular contour-oriented coding techniques are the derivative chain code, the polygonal approximation, and the geometrical curve approximation. The advantage of the derivative chain code algorithm is that it can efficiently and losslessly code *connected* contour arcs. The main drawback lies in the coding of the starting point of the contour arc. The two other techniques are lossy and their performances depend on the application. They are, in particular, difficult to use efficiently if the accepted loss in contour position is very low.

In the case of contour coding by motion compensation, the information to transmit is the prediction error. This kind of signal is generally made of a large number of small regions resulting in a large number of isolated contours. In this case, the cost of the starting points is high, and techniques such as the derivative chain code become less efficient. In the following, we will introduce a morphological tool for shape-oriented contour coding. It allows a flexible contour representation, which is efficient in the case of many separated contour arcs or of contours with few details.

B. Geodesic Skeleton

A solid theory for the Euclidean skeleton has been established in [19]. The skeleton $r(X)$ of an open set X is defined as the locus of the maximal open ball $\overset{\circ}{B}_\rho$ included in X . Each maximal ball is characterized by its center x and its radius ρ .

The mapping $x \rightarrow \rho, x \in r(X)$ is called the quench function of $r(X)$. In [38], the theory is developed into an algebraic and a topographic branch. Reference [38] refers to the first one by Lantuéjoul's formula, as follows:

$$r(X) = \bigcup_{\rho>0} r_\rho(X) = \bigcup_{\rho>0} [\epsilon^\rho(X) \setminus \gamma^1(\epsilon^\rho(X))] \quad (4)$$

where $\epsilon^\rho(X)$ is an erosion of size ρ , and $\gamma^1(X)$ stands for the opening of size one. The topological branch is relevant when preservation of connectivity is necessary. This occurs mainly in applications dealing with the topological study of objects (how many objects, how many holes, how many branches). In image coding, the goal is to represent objects with the lowest number of bits, and the connectivity preservation is not mandatory. According to [38], for such application it is necessary to find a skeleton decomposition requiring the three following points:

- 1) existence and unicity of $r(X)$ for a set X
- 2) equivalence between the set X and the reconstructed set X' from its skeleton representation
- 3) an explicit formula to compute the skeleton.

Furthermore, it is important that the explicit formula mentioned in point 3 produces a skeleton with as few points as possible to obtain a high compression of the contour information. In this section, the skeleton decomposition for contour coding will be the topic of our discussion. It will be possible to apply the results for the coding of contour residues.

The skeleton has been employed for coding of binary images [15]. It was shown that the skeleton decomposition using (4) contains redundant points that are not necessary for a perfect reconstruction. The skeleton, however, has never been applied to coding of segmented images containing several regions. The reason is perhaps that a direct application of (4) results in a redundant coding of contours. As a matter of fact, every contour belongs to two different neighboring regions and will be represented by the skeleton of each of the regions. Another drawback of 4 is that it is an iterative process that has a long computation time.

The Geodesic Skeleton Based on Distance Transformations: Reference [20] presents and develops some results established in [19], leading to a new definition for the skeleton on the digital grid. It characterizes the skeleton points as a particular set on the distance function, and leads to an extremely fast algorithm. In the following, we will give a short summary.

Given a set X , the distance function ρ at a point x of X is defined as

$$\rho_X(x) = d(x, X^c) = \inf_{y \notin X} d(x, y). \quad (5)$$

It can be easily seen that there exists always at least one point y_0 on the boundary of X such that $d(x, y_0) = \rho_X(x)$. Starting from a point x , we call *upstream* of x , the set of points y satisfying the relationship

$$\rho(y) = \rho(x) + d(x, y). \quad (6)$$

If the upstream of x consists of just x itself, then x is a skeleton point. It leads to the following definition.

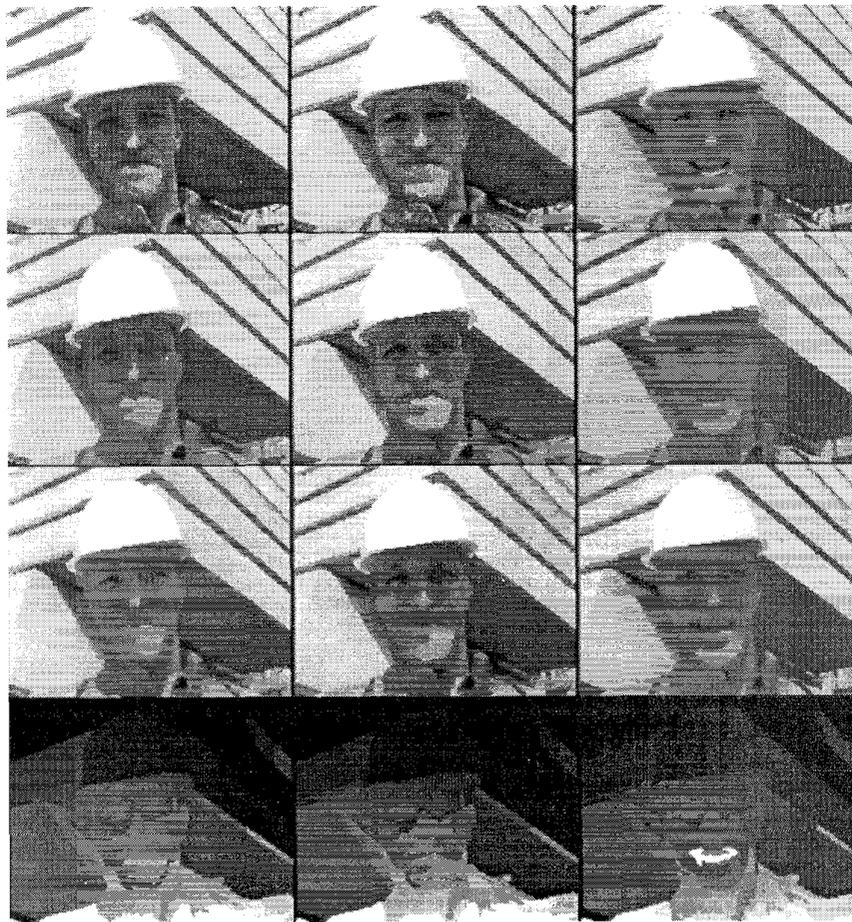


Fig. 11. Example of projection and decision with the region-growing version of the watershed. First row: original frames. Second row: projection results. Third row: coded image after projection and introduction of new regions. Fourth row: final segmentation.

Definition: The skeleton is the set of points without other upstream than themselves.

According to this definition, for each point x , all points $y \in X$ have to be checked to know whether an upstream of x exists, in which case x is not a skeleton point. However, in the digital case, it is sufficient to only consider all neighbors of x , which are defined by the graph used for the distance computation in (5) and (6) (see [2]). Frequently, the chessboard mask is used ($= 3 \times 3$ square). From the skeleton definition, which produces exactly the same skeleton as in (4), the geodesic skeleton can be derived. The idea is to exclude contours that are already known in the distance computation in (5). This means that the point y_0 is restricted to some specific boundary points of X . If we denote by Z the area that is already coded, we define a new reference by

$$K = Z \cup X \tag{7}$$

which will be used in the distance computation, and yields

$$\rho_K(x) = d(x, K^c) = \inf_{y \notin K} d(x, y) \quad \forall y \in E, x \in X. \tag{8}$$

The skeleton is, again, the points without other upstream than themselves, and is denoted $r_K(X)$. An illustrative example is

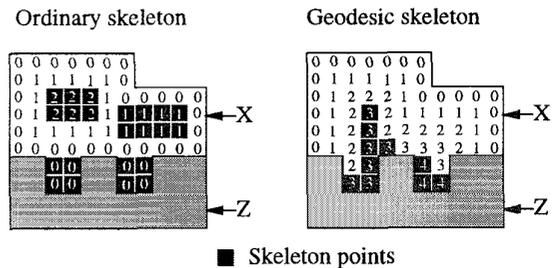


Fig. 12. Comparison between ordinary and geodesic skeletons based on distance functions.

given in Fig. 12, where the ordinary and the geodesic skeleton are shown for a region where another region below it has already been coded.

As shown in [15], the skeleton contains redundant points. It is also the case for the geodesic skeleton. These points can be removed while still allowing a lossless reconstruction. For their detection, the same approach as [15] is followed.

- 1) A redundancy matrix is computed. For every pixel, the number of structuring elements that cover it are counted.

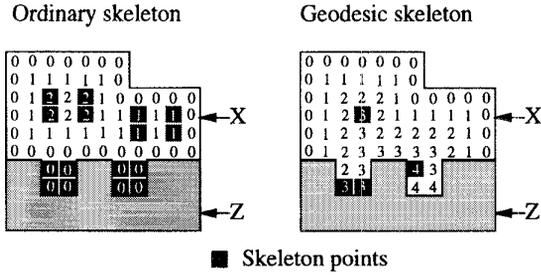


Fig. 13. Comparison between ordinary and geodesic skeletons without redundant points.

- 2) All skeleton points that cover an area that is already completely covered by other maximal structuring elements are removed.

This procedure requires dilation of each skeleton point by the corresponding quench function (point 1), and scanning of each dilated region for the presence of pixels coming from other skeleton points (point 2), which is extremely time consuming. Therefore, a particular dilation procedure for the skeleton points based on hierarchical queues is proposed, which allows a much faster implementation than when ordinary dilation techniques are used. The dilation procedure will be explained in the next subsection. Examples of skeletons without redundant points are shown in Fig. 13.

In [2], we demonstrate that the geodesic skeleton *always* contains an equal number of, or less, skeleton points than the ordinary skeleton. Experimental results confirm that a reduction of 50% of skeleton points is obtained on average for a segmented image [3]. The distance function can be very efficiently implemented in only two image scans, and a third image scan is necessary to detect all points without other upstream than themselves in the distance function.

Fast Reconstruction of the Skeleton Subset: In almost all image coding applications, the decoder should work in real time. This requirement implies a structure that is simple and efficient. The geodesic skeleton allows an exact reconstruction of the decomposed region by the following procedure:

$$X = \bigcup \{x \in r_K(X) : \delta_Z^{\rho_K(x)}(x)\} \quad (9)$$

where $\delta_Z^{\rho_K(x)}$ denotes geodesic dilation with respect to Z and of size $\rho_K(x)$ [44], [31].

Thus, a simple reconstruction based on unions of dilations is guaranteed. However, the reconstruction process as defined in (9) is not efficient. The skeleton guarantees that every inscribed disk covers at least one point that is not covered by any other disk, but many points will be covered by several disks. This implies that during the reconstruction by dilation procedure, many points will be processed several times. To avoid this and to provide a more efficient dilation algorithm, we propose the use of hierarchical queues. The image will be scanned only once to put all skeleton points into the queue, and only points involved in the dilation are processed.

Hierarchical queues have already been used for the watershed algorithm. Here, the priority of the queues is defined by

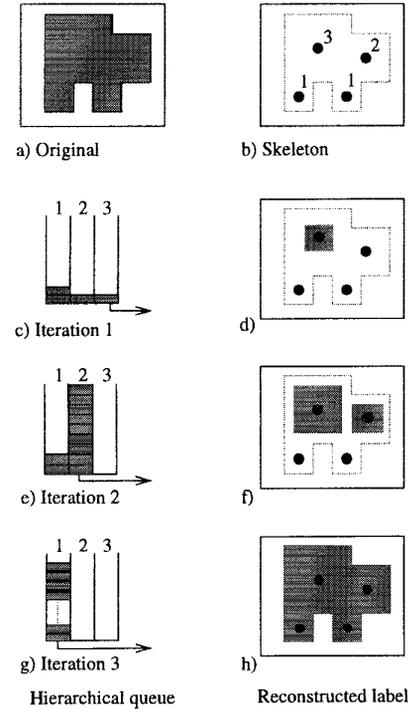


Fig. 14. Reconstruction using hierarchical queues.

the size of the quench function. The following reconstruction algorithm is proposed.

- 1) Put all skeleton points into a hierarchical queue at a priority corresponding to their quench function.
- 2) Retrieve all points from the nonempty queue of highest priority and dilate them by one unit.
- 3) Put all *new* points in the queue of the next lower priority.
- 4) Repeat points 2–3 until the hierarchical queue is empty and the label reconstructed.

Fig. 14 illustrates an example of the reconstruction procedure. Fig. 14(a) shows the original region. Its skeleton decomposition is shown in Fig. 14(b). It consists of one point with quench value three, one point with quench value two, and two points with quench value one. The image is initially scanned and the points put into the hierarchical queue. Thus, on the highest hierarchical level, there is one point. On level two, there is one point and, on level one, there are two points. See Fig. 14(c).

The process starts by extracting the first element out of the queue, i.e., the skeleton subset point with quench value three, and to dilate it by one unit as in Fig. 14(c) and (d). If all neighbors are within the reference set, then eight new points are created, each of which will have to be dilated by size two, and they are put into the queue at level two. Therefore, the queue at level two now contains nine elements; see Fig. 14(e). If queue number three is empty, dilation is continued at level two. Each element of queue two is taken out and dilated by one unit as in Fig. 14(f). The neighbors that have not yet been processed are put into queue at level one. When queue two is empty, queue one will contain 26 elements at most; see Fig. 14(g). They are extracted and dilated by one; see

Fig. 14(h). The iteration stops when queue one is empty and the region reconstructed.

C. Application of Geodesic Skeleton for Contour Coding

In the following, we assume that the partition is coded by motion compensation and that the *geodesic skeleton* representation is used to code the compensation error of the regions. An exact definition of the reference set used in the geodesic computation is required for an application of the geodesic skeleton for contour error coding. The reference set is the union of the regions that are known to the receiver. At a given time instant, these regions are composed of two sets: i) motion compensated regions; and ii) the already coded compensation error. Therefore, $Z = C \cup \hat{P}$ where C are the coded regions and \hat{P} is the predicted region from the previous frame. The skeleton is then computed using (8). It exhibits several properties, as follows:

- 1) the size of quench function is proportional to the reconstruction accuracy;
- 2) the skeleton "knows" if a residue has to be added or deleted from the prediction;
- 3) the skeleton points have a tendency to move toward known contours.

Size of Quench Function: Points with small quench function correspond to small associated geodesic structuring elements. Omitting skeleton points results in a contour simplification proportional to the size of the quench function. However, the simplification is reversible, since those points can be added back.

Adding/Deleting Prediction Residues: Two different kind of prediction residues are possible:

- 1) the residue has to be added to the prediction;
- 2) the residue has to be deleted from the prediction.

The skeleton decomposition automatically codes this information. Let us denote by X the region to be skeletonized, as in the previous section. By definition, $r_K(X) \subset X$. For residues to be added, $X = P \setminus \hat{P}$, where P denotes the real, current region and \hat{P} is the predicted region from previous frames. Therefore, $r_K(X) \notin \hat{P}$. Analogous, for residues to be deleted, $X = \hat{P} \setminus P$ and $r_K(X) \in \hat{P}$. Thus, a simple inspection whether the skeleton point lies in the predicted region indicates how the residue has to be reconstructed.

Spatial Location of Skeleton Points: The quench function and the spatial location of the skeleton points have to be coded for transmission. The former can be entropy coded directly. The latter can rely on run-length coding. However, because the skeleton points are sparsely distributed in the image, the corresponding runs are extremely long. The entropy of each run is either high or a run has to be broken up into many shorter pieces.

A remedy to the problem is the fact that geodesic skeleton points have a tendency to move toward known contours, as hinted by Fig. 12. This assumption is confirmed by experimental results. In the graph of Fig. 15, the distance of skeleton points to known contours is plotted. Roughly 80% of all points are direct neighbors of known contours. Therefore, the skeleton points of distance one are included in the subset

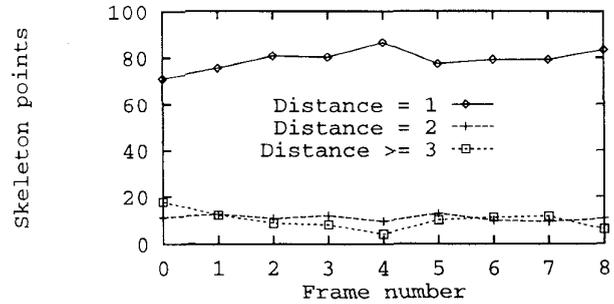


Fig. 15. Evolution of the distance of skeleton points to known contours.

TABLE III
BITSTREAM COMPARISON FOR CONTOUR
INFORMATION OF QCIF IMAGE SEQUENCES

	MissAmerica	CarPhone	Foreman
Chain Intra	20.4 kBits/s	23.7 kBits/s	22.5 kBits/s
Chain Residues	7.6 kBits/s	13.9 kBits/s	17.5 kBits/s
Skeleton Residues	6.2 kBits/s	11.5 kBits/s	17.8 kBits/s

$\delta^1(\hat{P}) \setminus \hat{P}$ or $\hat{P} \setminus \epsilon^1(\hat{P})$, according to whether a residue has to be added or deleted. Run-length coding should be performed only on this subset, resulting in a great reduction of the average length of the runs. A special symbol is introduced in the bitstream to indicate that all points of distance one have been coded. Next, skeleton points at a distance of two are coded. They are all situated on the subset $\delta^2(\hat{P}) \setminus \delta^1(\hat{P})$ or $\epsilon^1(\hat{P}) \setminus \epsilon^2(\hat{P})$ and runs are again defined on this subset. The remaining points are coded the same way. This procedure allows a reduction in bit rate of about 20%.

Other encoding strategies are possible. For example, the skeleton points can be approximated by lines and curves and then chain-coded. The technique gives interesting results in the case of redundant skeletons with mostly connected points [1]; however, our experiments have shown that coding of nonredundant skeletons provides better results.

D. Coding Performances

Coding results are illustrated in Fig. 16. The prediction errors are simplified by morphological filters and then coded and reconstructed [2]. In Table III, results are given for the bit requirements for contour information over 150 frames. The skeleton is compared to the chain-code method, known as one of the best contour coding techniques. Chain code is applied in two ways: In an intramode on each frame individually without exploiting the temporal correlation [17], and on the prediction errors the same way as the skeleton [9], [35]. In the first case, a lossless coding is performed, but at a much higher bit rate, which does not justify the marginal improvement in image quality. In the second case, chain code produces the same contour quality as the skeleton, at a roughly 10–15% higher bit rate in general. The reason is that there are many isolated and short-contour arcs that are expensive to code, because it is necessary to code the starting point and the ending point of the chain. The skeleton copes better with this kind of contour images. For the Foreman sequence, long chains of

TABLE IV
TOTAL BITSTREAM BY THE SKELETON DECOMPOSITION
(QCIF IMAGES, 150 FRAMES, 5 HZ FRAME-RATE)

	MissAmerica	CarPhone	Foreman
Motion	1.1 kBits/s	1.4 kBits/s	2.1 kBits/s
Texture	15.3 kBits/s	16.6 kBits/s	19.9 kBits/s
Contour	6.2 kBits/s	11.5 kBits/s	17.8 kBits/s
TOTAL	22.6 kBits/s	29.5 kBits/s	40.0 kBits/s

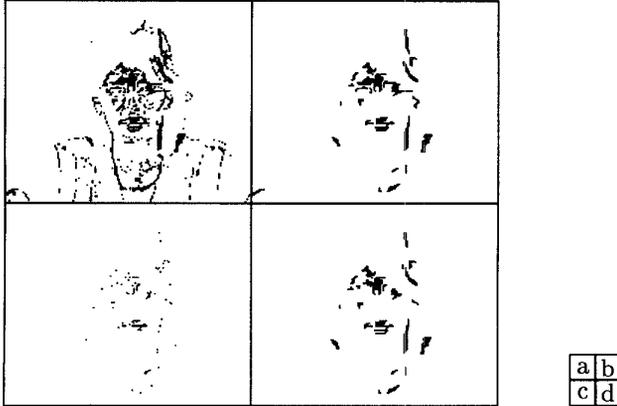


Fig. 16. Contour coding with geodesic skeleton: a) Prediction residues; b) simplified residues; c) skeleton points; d) reconstructed residues.

contour estimation errors make the chain-code and skeleton techniques about equally performing. Results for the total bitstream requirements include bits for motion information, texture information [35], and contour information and are shown in Table IV.

VI. MORPHOLOGICAL INTERPOLATION AND TEXTURE CODING

A. Classical Approach to Interpolation

Interpolative coding techniques [25] are based on the coding and transmission of a subset of pixels of the original image, so that, on the receiver side, the remaining pixels have to be interpolated from the transmitted information alone. The reconstructed image is approximated by continuous functions with some permissible error at the interpolated pixels.

Examples of the application of interpolative coding techniques can be found in [4] and [29], where perceptual considerations about the human visual system lead to a model of the image for coding purposes based on the concept of the “raw primal sketch.” In such model, the image is assumed to be made mainly of areas of smoothly changing intensity separated by discontinuities or edges. Under this assumption, it is possible to obtain a reconstruction of the original image only from the information of the geometric structure of the discontinuities and the amplitudes at the edge pixels (sketch data).

In “sketch-based” coding methods, the selection of the sketch data is the key issue for the coding process. For a given interpolation strategy, it sets the trade-off between the coding cost and the quality of the reconstructed image. Furthermore,

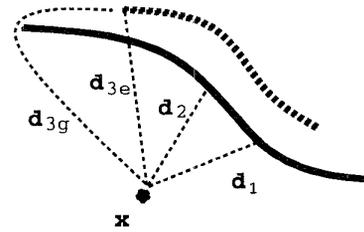


Fig. 17. Geodesic distance measure for the interpolation of pixel x .

interpolation algorithms for the reconstruction of the image from sketch data present a main drawback: They are usually based on filtering processes² with rather high computational load.

In this section, a morphological tool intended to solve the 2-D interpolation problem is presented. It is based on geodesic dilation, resulting in a highly efficient process compared to those that perform linear filtering of the initial image. The proposed algorithm is general in the sense that it is able to perform interpolation from any set of given pixels. To illustrate this point, two different coding strategies will be shown that validate the use of morphological interpolation as an efficient method for texture coding, regardless of its use within a region-based framework or in a block-based approach. A third strategy specifically devoted to region-based schemes can be found in [33].

B. Morphological Interpolation

The target of the morphological interpolation algorithm is to approximate the amplitudes of unknown pixels of the image by fitting a surface on a subset of pixels of known values, called *initial pixels* in the following. Such surface is constrained to be maximally smooth between the known pixels. The smoothness condition is stated in the sense that pixel-to-pixel variations should be minimized. This would lead to straight-line interpolation between two initial values for 1-D signals.

A suitable strategy for the 2-D interpolation is to compute at each point the average of the amplitudes of the initial pixels weighted by the inverse of the distances to each of them [40]. The nearest pixels have stronger influence than the distant ones, and the interpolated amplitudes change slowly in the areas in between. The distance measure is taken as the *geodesic distance* within the set of unknown pixels. The advantage of using the geodesic distance is illustrated in Fig. 17. The set of initial pixels for the interpolation is indicated by thick solid and dashed lines. Let us suppose that the dashed line represents the upper edge of a spatial transition and the solid line represents the lower edge of the transition. The influence of the amplitude values of the upper edge (dashed line) at pixel x will be given by the inverse of the geodesic distance d_{3g} , which is larger than the Euclidean distance d_{3e} . Therefore, the interpolated values at pixel x will be mainly influenced by the initial pixels of the solid line. As a result, the use of the geodesic distance allows the preservation of the transitions.

²In [4] and [29] these are iterative overrelaxation processes governed by the heat diffusion equation.

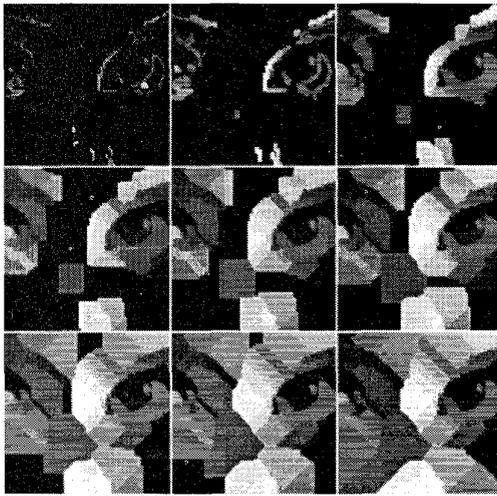


Fig. 18. Geodesic propagation process from the set of initial pixels.

The interpolation strategy described above may be efficiently implemented by an iterative algorithm based on morphological operators [5]. Each iteration of the algorithm performs a *geodesic propagation* followed by a *smoothing step*. Instead of computing geodesic distances from all the unknown pixels to every initial pixel, the amplitude values of the initial pixels are propagated by geodesic dilation to fill the empty areas of the image and then, the transitions appearing outside the set of initial pixels are smoothed. Fig. 18 shows some intermediate images corresponding to the first propagation process from the set of initial pixels. When two or more propagation fronts originated from the initial pixels of different amplitudes meet, the process stops and a false transition is created. Pixels on both sides of the false transitions compose the set of *secondary* pixels. A gray-level value equal to the average of the gray-level values on both sides of the transition is assigned to each secondary pixel. This is the smoothing step.

Then, the second iteration is performed: The propagation step propagates the gray-level values from the sets of initial as well as secondary pixels. The propagation creates new false transitions that define a new set of secondary pixels where gray-level values are smoothed. Note that this new set of secondary pixels generally does not include the first set of secondary pixels. This process of propagation of values from the initial and secondary pixels and smoothing of the gray levels of the new secondary pixels is iterated until idempotence.

Fig. 19 shows the initial and interpolated images for the second, fourth, and eighth iteration of the interpolation algorithm. The middle column gives the sets of initial and secondary pixels and the right column shows the propagation results. Observe the iterative smoothing of the false transitions.

The two steps of the iterative morphological interpolation process can be efficiently implemented. The geodesic propagation is implemented by means of a FIFO queue. First, the initial image is scanned in order to find all the "empty" neighbors of the initial pixels (at geodesic distance 1). Each

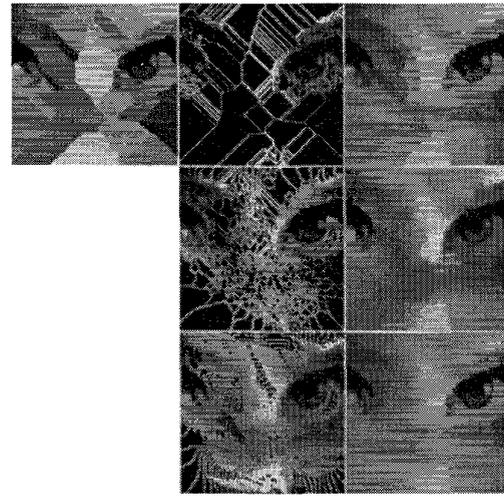


Fig. 19. Three iterations (second, fourth, and eighth) of the geodesic propagation process. Middle column: initial and secondary pixels. Right column: propagation result.

TABLE V
EXECUTION TIMES OF MORPHOLOGICAL INTERPOLATION
VERSUS INTERPOLATION BASED ON DIFFUSION EQUATIONS

Interpolation technique	linear diffusion	multigrid diffusion	morphological
Execution time	458,3 s.	60,3 s.	2,4 s.
No. of iterations	2980 it.	equivalent to 376 it.	13 it.

one of these pixels is given the gray-level value of its initial pixel neighbor and then its location is put into the queue. If the pixel happens to have more than one neighboring initial pixel, its amplitude is chosen randomly among them. During the propagation, one pixel is extracted from the queue and its amplitude is propagated to all its empty neighbors whose locations, in turn, are put in the queue. The process stops when the queue is empty. Therefore, each pixel is treated only once to perform a complete geodesic propagation.

As far as the smoothing step is concerned, the secondary pixels are extracted by means of a morphological Laplacian operator, which is defined as the difference between the gradients by dilation and by erosion, as follows:

$$L(f) = g^+(f) - g^-(f) = \delta^1(f) + \epsilon^1(f) - 2f. \quad (10)$$

This is a nonlinear approximation to the Laplace operator in continuous space that was first studied in [41] for edge detection. It is greater than zero at the lower edge of the transitions and smaller than zero at the upper edge. In flat surfaces or slanted planes without convexity changes, it cancels out. Therefore, the morphological Laplacian can be used as a transition detector. Moreover, if it is divided by two and added back to the result of the propagation, the pixels at the false transitions are smoothed.

The efficiency of the morphological interpolation algorithm in terms of computational load is illustrated in Table V. Comparative figures of execution time³ for this operation and for an interpolation by iterative overrelaxation governed by

³CPU times were computed on a Sun SPARC10 workstation.

TABLE VI
COMPRESSION RATIO FOR THE EXAMPLE OF
CODING BY MAXIMUM AND MINIMUM POINTS

Iteration	1	10	25	50	75	100
No. of points	10	55	93	183	270	382
Compression	451	125	75	40	30	20

the heat diffusion equation [4] are reported for the example shown in Figs. 18 and 19. The interpolated result is almost indistinguishable in both cases. Please note the drastic reduction in the number of iterations needed for the morphological interpolation technique. Each pixel of the image to interpolate is treated hundreds of times less. Furthermore, each iteration of the morphological interpolation does not require any multiplication, which decreases the time of each individual iteration compared to the linear filtering technique. This explains the reduced execution time of the described nonlinear interpolation process. Clearly, there is no need for multigrid techniques for speeding up convergence when dealing with sparse sketch data if the morphological interpolation algorithm is used.

C. Application of Morphological Interpolation for Texture Coding

The application of morphological interpolation for texture coding relies on the selection of a set of initial pixels. This set of pixels should at the same time allow a good restoration of the image by interpolation, and lead to a compact representation. In the following, two texture-coding strategies are presented. In the first one, the set of initial pixels is composed of lines of maximum and minimum curvatures, whereas the second one involves the points of local maximum and minimum.

Coding by Maximum and Minimum Curvature Lines: The first example involves an image representation based on networks of lines. Networks of lines are interesting for coding because they can be efficiently coded using derivative chain-code techniques. The main difficulty consists in selecting lines that can lead to a good restoration of the image. A first idea may be to select the crest lines of maximum and minimum, that is, the watershed of the image and of its dual. However, some experiments have shown that the resulting interpolated image lacks transitions and that much better results can be obtained by using the lines of maximum and minimum curvatures. These lines may characterize maxima and minima of the signal as well as its transitions.

The maximum curvature lines are extracted in three steps: first, the image is prefiltered by a dynamics filter [8] that removes low-contrasted components while preserving the gray-level shape of the remaining components (unlike the h -max, which does not preserve the gray-level shape of the signal but only its contour). Second, the morphological Laplacian is computed in order to achieve edge enhancement. Indeed, it can be shown that the morphological Laplacian is an approximation of the second derivative, so that the extrema of the Laplacian image correspond to those of the signal second derivative. Third, the watershed of the Laplacian extracts a



Fig. 20. Morphological interpolation from lines of maximum and minimum curvature. First row: original and prefiltered image. Second row: lines of maximum (white) and minimum (black) curvature, interpolated image (compression ratio 30).

network of lines identifying the lines of maximum curvature. The dual process extracts the lines of minimum curvature.

This process is illustrated in Fig. 20. The upper row presents the original image and the prefiltering. The lower row gives the network of maximum curvature lines (in white) and of minimum curvature lines (in black). The position of the networks' pixels and the gray-level values of these pixels in the original image have to be coded and transmitted to the receiver. Since the network is composed of connected pixels, a derivative chain code is used to code the pixels' positions. The gray-level values are coded by polynomial approximation. More precisely, the network is broken at each triple point (points with more than two branches) and the gray level values of the resulting curves are approximated by a second order polynomial. The resulting three coefficients are quantized, entropy coded and transmitted. At the receiver, the networks of lines with the approximated gray levels are used as set of initial pixels for the interpolation. The restored image is presented on the right side of the lower row of Fig. 20. As can be seen, most of the important smooth variations, as well as transitions of the signal, have been coded. The compression ratio achieved with this strategy is equal to 30.

Coding by Local Maximum and Minimum Points: The second example deals with an image representation involving isolated points. Here, the objective is to select the smallest number of points leading to a good restoration of the image. One possible solution consists of using an iterative selection process.

In the first iteration, the pixels of absolute maximum and minimum amplitudes are selected. A first reconstruction by interpolation is performed, and the residue with the original frame is computed. From this residue, a second set of maximum and minimum points are selected. They are used together with the first set of pixels to compute a second restoration. This process is iterated in order to reach a sufficient quality of the restored image. Fig. 21 illustrates various iteration steps. The numbers of iterations are, respectively, of 1, 10, 25, 50, 75, and 100. For each iteration, the initial pixels, the

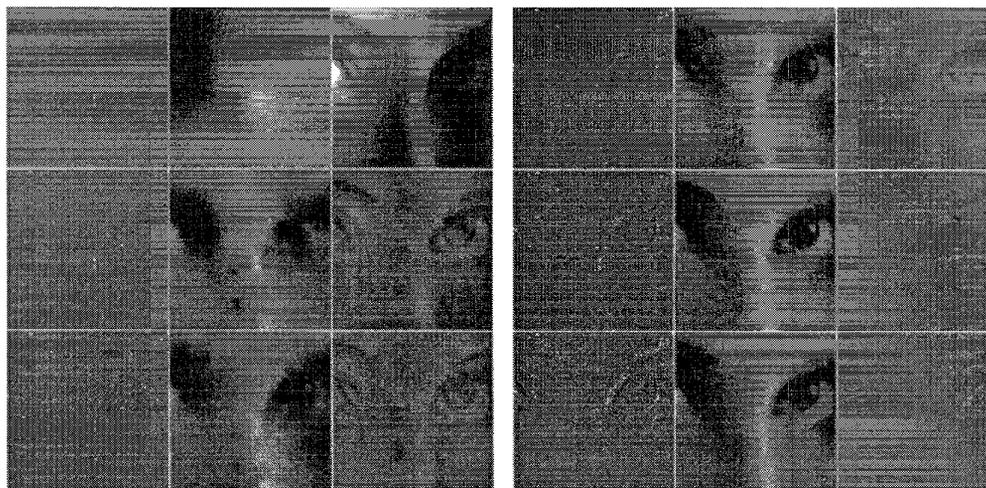


Fig. 21. Morphological interpolation from a set of isolated points. For each row: set of initial points, interpolated image, residue. The compression ratios are 451, 125, 75, 40, 30, and 20.

interpolated image, and the residue are shown. It can be seen how isolated points are progressively introduced and how interpolated image quality is improved.

The coding of the isolated points position is achieved by an Elias code [6]. The gray levels are simply stored in a buffer following a scanning order, and are entropy coded. Table VI gives the number of isolated points together with the compression ratio.

VII. CONCLUSIONS

In this paper, the usefulness of some morphological tools for image and video compression has been presented and discussed. Four sets of morphological transformations have been reviewed: *connected operators*, *the region-growing version of the watershed*, *the geodesic skeleton*, and *a morphological interpolation technique*.

- *Connected operators* solve the problem of image simplification while preserving the contour information. This very important feature relies on the separation of a binary selection step and a reconstruction step. The selection step decides whether or not a connected component or a flat zone has to be preserved. The reconstruction process defines the shape of the selected components. Several simplification criteria can be obtained, the most popular ones being size-oriented or contrast-oriented. These operators are also attractive because they can be very efficiently implemented by using FIFO queues. For coding applications, they are useful for most of the simplification steps, but especially for the segmentation. Moreover, the concept of flat zones is very useful for the feature extraction step of the segmentation.
- *The region-growing version of the watershed* has two main advantages. First, it allows the direct processing of the signal to segment. The loss in contour precision implied by the use of the gradient is avoided. Note that, although this loss is limited to two pixels in the case of

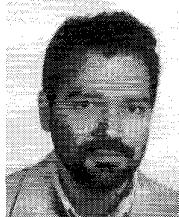
still images, it can become very large for moving images. Second, this version of the watershed allows the introduction of complex criteria, taking into account the gray-level homogeneity as well as the contour complexity, that is, the contour coding cost. This version of the watershed can also be very efficiently implemented by hierarchical queues. For coding applications, this watershed algorithm is particularly suitable for the precise contour definition (decision step) and for the temporal linking of regions (projection step).

- The *geodesic skeleton* has been proposed to code the contour prediction error within a motion-compensated coding strategy. It avoids coding twice each contour, and leads to a flexible representation of the contour information. Here also, hierarchical queues turn out to be very useful for the efficient implementation of the technique.
- Finally, *morphological interpolation* is a very efficient tool for interpolation on nonregular grids. It allows the restoration of an image from a reduced number of points. Moreover, it gives a large amount of freedom in the selection of the initial points. Two examples have been shown to illustrate this feature, one involving a network of lines and another using isolated points. These texture-coding strategies can be used within a region-based or a block-based approach. As before, queues turned out to be crucial elements for the efficient implementation of the algorithm.

REFERENCES

- [1] J. W. Brandt, A. K. Jain, and V. R. Algazi, "Medial axis representation and encoding of scanned documents," *J. Vis. Commun. Image Repres.*, vol. 2, pp. 151–165, June 1991.
- [2] P. Brigger, S. Ayer, and M. Kunt, "Morphological shape representation of segmented images based on temporally modeled motion vectors," in *Proc. IEEE Int. Conf. Image Processing*, Austin, Texas, Nov. 1994, vol. III, pp. 756–760.

- [3] P. Brügger, F. Meyer, and M. Kunt, "The geodesic morphological skeleton and its fast reconstruction," in J. Serra and P. Soille, Eds., *Second Workshop Math. Morphol. Applic. Signal Processing*. Boston: Kluwer, pp. 133–140.
- [4] S. Carlsson, "Sketch based coding of grey level images," *EURASIP Signal Processing*, vol. 15, no. 1, pp. 57–83, July 1988.
- [5] J. R. Casas and L. Torres, "Feature-based video coding using mathematical morphology," in *Proc. EUSIPCO 94, VII Europ. Signal Processing Conf.* Edinburgh, U.K., Sept. 1994, pp. 143–146.
- [6] P. Elias, "Predictive coding—part I," *IRE Trans. Inform. Theory*, vol. IT2, pp. 16–33, Mar. 1955.
- [7] D. Florencio and R. Schafer, "Critical morphological sampling and applications to image coding," in J. Serra and P. Soille, Eds., *Second Workshop on Mathematical Morphology and its Applications to Image Processing*. Boston: Kluwer, 1994, pp. 109–116.
- [8] M. Grimaud, "A new measure of contrast: the dynamics," in *Proc. Vis. Commun. Image Processing'92*, San Diego, CA, July 1992, vol. 1769, pp. 292–305.
- [9] C. Gu and M. Kunt, "Contour simplification and motion compensated coding," to be published in *EURASIP, Signal Processing: Image Commun.*, 1995.
- [10] H. Heijmans, "Theoretical aspects of gray level morphology," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 6, pp. 568–592, 1991.
- [11] J. C. Klein, "Conception et réalisation d'une unité logique pour l'analyse quantitative d'images," Ph.D. dissertation, Nancy University, France, 1976.
- [12] R. Kresch and D. Malah, "Morphological reduction of skeleton redundancy," J. Serra and P. Salembier, Eds., in *Proc. First Workshop Math. Morphol. Applic. Signal Processing*, Barcelona, Spain, May 1993, pp. 145–150.
- [13] M. Kunt, M. Bernard, and R. Leonardi, "Recent results in high compression image coding," *IEEE Trans. Circuits Syst.*, vol. 34, no. 11, pp. 1306–1336, Nov. 1987.
- [14] M. Kunt, A. Ikononopoulos, and M. Kocher, "Second generation image coding techniques," *Proc. IEEE*, vol. 73, no. 4, pp. 549–575, Apr. 1985.
- [15] P. A. Maragos and R. W. Schafer, "Morphological skeleton representation and coding of binary images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 5, pp. 1228–1244, Oct. 1986.
- [16] ———, "Morphological filters part I: Their set-theoretic analysis and relations to linear shift-invariant filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 8, pp. 1153–1169, Aug. 1987.
- [17] F. Marqués, J. Sauleda, and T. Gasull, "Shape and location coding for contour images," in *Proc. Picture Coding Symp.*, Lausanne, Switzerland, Mar. 1993, pp. 18.6.1–18.6.2.
- [18] F. Marqués, V. Vera, and A. Gasull, "Recursive image sequence segmentation by hierarchical models," in *Proc. 12th Int. Conf. Pattern Recog.*, Jerusalem, Israel, Oct. 9–13, 1994, pp. 523–525.
- [19] G. Matheron, "Examples of topological properties of skeletons," in J. Serra, Ed., *Image Analysis and Mathematical Morphology, Vol. 2: Theoretical Advances*. New York: Academic, 1988, ch. 11, pp. 217–238.
- [20] F. Meyer, "Skeletons and watershed lines in digital spaces," *SPIE*, vol. 1350, pp. 85–102, 1990.
- [21] ———, "Color image segmentation," in *Proc. 4th Int. Conf. Image Processing Applic.*, Maastricht, The Netherlands, May 1992, pp. 303–304.
- [22] ———, "Morphological image segmentation for coding," in J. Serra and P. Salembier, Eds., *First Workshop Math. Morph. Applic. Signal Processing*, Barcelona, Spain, May 1993, pp. 46–51.
- [23] F. Meyer and S. Beucher, "Morphological segmentation," *J. Visual Commun. Image Represent.*, vol. 1, no. 1, pp. 21–46, Sept. 1990.
- [24] H. G. Musmann, M. Hotter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Processing, Image Commun.*, vol. 1, no. 2, pp. 117–138, Oct. 1989.
- [25] A. N. Netravali and J. O. Limb, "Picture coding: a review," *Proc. IEEE*, vol. 68, no. 3, pp. 366–406, Mar. 1980.
- [26] M. Pardàs and P. Salembier, "3D morphological segmentation and motion estimation for image sequences," *EURASIP Signal Processing*, vol. 38, no. 2, pp. 31–43, Sept. 1994.
- [27] ———, "Time-recursive segmentation of image sequences," in *Proc. EUSIPCO 94, VII Europ. Signal Processing Conf.*, Edinburgh, U.K., Sept. 13–16, 1994, pp. 18–21.
- [28] S. Rajala, M. Civanlar, and W. Lee, "Video data compression using three-dimensional segmentation based on HVS properties," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, New York, 1988, pp. 1092–1095.
- [29] X. Ran and N. Farvardin, "A perceptually motivated three-component image model, part I: description of the model; part II: application to image compression," *IEEE Trans. Image Processing*, vol. 4, no. 4, pp. 401–415 and 430–447, Apr. 1995.
- [30] P. Salembier, "Multi-criterion segmentation for image coding," in J. Serra and P. Salembier, Eds., *First Workshop Math. Morph. Applic. Signal Processing*, Barcelona, Spain, May 1993, pp. 40–45.
- [31] ———, "Morphological multiscale segmentation for image coding," *EURASIP Signal Processing*, vol. 38, no. 3, pp. 359–386, Sept. 1994.
- [32] P. Salembier and M. Pardàs, "Hierarchical morphological segmentation for image sequence coding," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 639–651, Sept. 1994.
- [33] P. Salembier and R. Rué, "Texture coding using morphological interpolation," in *Proc. 1995 IEEE Workshop Nonlinear Signal Image Processing*, Halkidiki, Greece, June 20–22, 1995, pp. 258–261.
- [34] P. Salembier and J. Serra, "Flat zones filtering, connected operators and filters by reconstruction," *IEEE Trans. Image Processing*, vol. 3, no. 8, pp. 1153–1160, Aug. 1995.
- [35] P. Salembier, L. Torres, F. Meyer, and C. Gu, "Region-based video coding using mathematical morphology," *Proc. IEEE*, vol. 83, no. 6, pp. 843–857, June 1995.
- [36] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.
- [37] ———, *Image Analysis and Mathematical Morphology, Vol II: Theoretical Advances*. New York: Academic, 1988.
- [38] ———, "Skeleton decompositions," *Image Algebra Morph. Image Processing III*, vol. 1769, pp. 376–386, 1992.
- [39] J. Serra and P. Salembier, "Connected operators and pyramids," *Image Algebra Math. Morph.*, vol. 2030, pages 65–76, July 1993.
- [40] P. Soille, "Spatial distributions from contour lines: an efficient methodology based on distance transformations," *J. Visual Commun. Image Representation*, vol. 2, no. 2, pp. 138–150, June 1991.
- [41] L. J. van Vliet, I. T. Young, and G. L. Beckers, "A nonlinear laplace operator as edge detector in noisy images," *Comput. Vision, Graphics, Image Processing*, vol. 45, pp. 167–195, 1989.
- [42] L. Vincent, "Graphs and mathematical morphology," *EURASIP Signal Processing*, vol. 16, no. 4, pp. 365–388, Apr. 1989.
- [43] ———, "Grayscale area openings and closings, their efficient implementation and applications," J. Serra and P. Salembier, Eds., in *Proc. First Workshop Math. Morph. Applic. Signal Processing*, Barcelona, Spain, May 1993, pp. 22–27.
- [44] ———, "Morphological gray scale reconstruction in image analysis: applications and efficient algorithms," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 176–201, Apr. 1993.
- [45] P. Willemin, T. Reed, and M. Kunt, "Image sequence coding by split and merge," *IEEE Trans. Commun.*, vol. 39, no. 12, pp. 1845–1855, 1991.



Philippe Salembier received a degree from the Ecole Polytechnique, Paris, France, in 1983, a degree from the Ecole Nationale Supérieure des Télécommunications, Paris, France, in 1985 and the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL) in 1991.

He was a Post-Doctoral Fellow at the Harvard Robotics Laboratory, Cambridge, MA, in 1991. From 1985 to 1989, he worked at Laboratoires d'Electronique Philips, Limeil-Brevannes, France, in the fields of digital communications and signal processing for HDTV. In 1989, he joined the Swiss Federal Institute of Technology in Lausanne, Switzerland, to work on image processing. At the end of 1991, after a stay at Harvard, he joined the Polytechnic University of Catalonia, Barcelona, Spain, where he teaches digital signal and image processing. His current research interests include image and sequence coding, image modeling, segmentation problems, texture analysis, mathematical morphology, and nonlinear filtering.



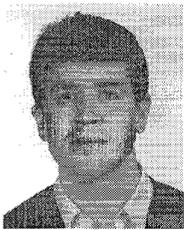
Patrick Brigger was born in Lucerne, Switzerland, on May 5, 1966. He received the degree of Electrical Engineer from the Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland, in 1992. Based on his diploma research, he was awarded the Jean Landry prize. In April 1992, he joined the Signal Processing Laboratory of the same institution (LTS) as a Ph.D. student.

From 1992–1993, he worked on an image analysis project related to the morphological description of cell cultures in collaboration with the University of Lausanne. In 1993, he joined the European Race project Morphoco, assuming the responsibilities within LTS. The principal objective of this work is related to contour coding using the skeleton decomposition. His major interests are in the development and application of morphological operators for image coding and biomedical image analysis.



Montse Pardàs received the M.S. degree in telecommunications and the Ph.D. degree from the Polytechnic University of Catalonia, Barcelona, Spain, in 1991 and 1995, respectively.

Her main research activity deals with nonlinear signal processing with a special emphasis on mathematical morphology. She is particularly interested in analysis problems for sequences such as segmentation, motion estimation, and coding.



Josep R. Casas received the Telecommunication Engineer and the Ph.D. degrees from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in 1990 and 1996, respectively.

He is Assistant Professor of television systems and image transmission courses at the UPC. His research interests are in image and video coding, morphological image processing, and advanced television systems.