

# ADAPTIVE CROSSING NUMBERS AND THEIR APPLICATION TO BINARY DOWNSAMPLING

ETIENNE DECENCIÈRE AND MICHEL BILODEAU

Centre de Morphologie Mathématique, Ecole des Mines de Paris, 35, rue Saint Honoré, 77305 Fontainebleau CEDEX, France

e-mail: Etienne.Decenciere@ensmp.fr

(Accepted May 21, 2007)

## ABSTRACT

A downsampling method for binary images is presented, which aims at preserving the topology of the image. It uses a general reference sampling structure. The reference image is computed through the analysis of the connected components of the neighbourhood of each pixel. The resulting downsampling operator is auto-dual, which ensures that white and black structures are treated in the same way. Experiments show, by visual inspection on the displayed images, that the image topology is indeed preserved satisfactorily.

Keywords: binary downsampling, digital topology, reference downsampling.

## INTRODUCTION

In this era of expanding mobile multimedia devices, small screens will soon be in every pocket. Their relatively small resolutions (the screen of a Personal Digital Assistant (PDA) is typically 320 by 320 pixels) pose display problems, worsened by the fact that visual digital documents are often thought for high resolution displays. For example, how can a faxed document, or a tourist brochure, scanned with a 200 dpi resolution, be conveniently displayed on a PDA screen?

As it can be seen, we are confronted with a severe downsampling problem. Moreover, these images often are binary or nearly so, like faxes, diagrams, maps, etc. In these particular cases, classical downsampling methods work very badly, because they aim at removing from the image those structures which cannot be represented at a lower resolution level. For example, depict a thin black line on a white background. If downsampled with a classical linear method (*i.e.*, high frequencies are filtered out before downsampling), this line will be smoothed away. If we require that the resulting image is binary, thin structures might be simply erased. In many application domains this is a normal, and welcome, feature. However, when displaying graphical data on small displays, the opposite might be more interesting, that is, preserving small structures when there is enough place in the image. In the case of binary images, this constraint can be expressed in mathematical terms as a homotopy preservation property.

This application was the initial motivation for our work, which explains some of the choices made during the study. However, very similar problems can be

found in other application domains. The following ones can be cited:

- multi-resolution representation of binary shapes for pattern recognition, a problem which has been studied by Borgefors *et al.* (1996; 1999; 2001);
- multi-resolution display of labelled images.

After this introduction, we will define the framework, and review the existing methods. Then, in section “Reference downsampling”, we will introduce a general adaptive downsampling scheme which will be used as basis in the following section for a binary downsampling method which aims at preserving homotopy. In the next section the results are presented and commented. Finally, conclusions are drawn.

Note that a first, shorter version of this paper was presented in the International Symposium for Mathematical Morphology (Decencièere and Bilodeau, 2005).

## FRAMEWORK AND OBJECTIVES

Only binary 2D images will be considered in this paper. They typically correspond to text, diagrams, graphics, or maps.

Thin and small structures in binary images are often semantically very important. Therefore, we want to preserve them through the downsampling procedure as long as possible. Borrowing vocabulary from the image compression world, we could say that we want to achieve *graceful degradation* of the information. Here, information will be of topological nature.

Of course, the detection of what is important is not trivial, nor it is easy to know how long it is possible to preserve data which is considered meaningful along several downsampling steps. We have made the hypothesis that the image topology is closely related to the correct perception of the binary image. Therefore, our goal is to produce a downsampling operator that preserves the image topology, when possible. Indeed, it is evident that in many cases, when resolution decreases, the resulting downsampled image cannot be homotopic to the original image. For example, a checkerboard image, where each pixel corresponds to one square, cannot be downsampled homotopically. However, in many other cases we believe that a topological approach might give interesting results.

When analysing binary images from a topological point of view, to avoid problems that will be seen later, in practice one often treats differently the “object” pixels and the “background” pixels. In our framework, we do not know beforehand if the important structures of a binary image are black or white. Therefore, we will treat them in the same way. In other words, the downsampling method should be *auto-dual*. Another reason for the adoption of this hypothesis is our wish to extend these results to gray scale images, where making a difference between “objects” and “background” is often impossible.

## STATE OF THE ART

The classical linear downsampling approach is based on the removal from the original image of those frequencies which are too high to be represented at a lower resolution level. They can be adapted to our framework by applying a convenient threshold after downsampling, in order to recover a binary image. The resulting downsampling operator can be auto-dual, however, preserving topological properties this way is not straightforward, as it will be shown in section “Results”.

Morphological downsampling methods are also based on the same idea (Haralick *et al.*, 1989; Heijmans and Toet, 1991; Florêncio and Schafer, 1994): first, they remove those structures which are considered too small to be represented at a lower resolution level, and then a point downsampling is applied. Clearly, this kind of approach is not adapted to our application.

In a series of articles, Borgefors *et al.* (1996; 1999; 2001) propose a multiscale representation of binary images. Their aim is to preserve the shape of the objects. Even if these methods tend to preserve the topology of the image, this is not their main objective.

Furthermore, the proposed downsampling methods are not auto-dual, an essential property in our framework.

Adaptive downsampling methods analyse the image contents before downsampling in order to preserve meaningful details when possible. A method based on the morphological tophat transformation has been proposed for downsampling grey level and binary images (Decencièrre *et al.*, 2000; 2001). It takes into account the size of the structures, by comparison with a structuring element (*i.e.*, a reference set), in order to favour those pixels which are considered more interesting. In this paper, we will adapt this approach to the case of binary images but, instead of geometric information, topological information will be used.

## REFERENCE DOWNSAMPLING

A general reference downsampling method has been introduced by Decencièrre *et al.* (2000; 2001). We present below a version adapted to binary images.

A binary image  $I$  is a binary function of  $\mathbb{Z}^2$ :

$$I: \quad \mathbb{Z}^2 \longrightarrow \{0, 1\} \\ (x, y) \longmapsto I(x, y).$$

The set of binary images is denoted  $\mathcal{I}$ . In the following, an *image* will mean a *binary image*. We will often identify an image  $I$  to the set  $\{p \in \mathbb{Z}^2 \mid I(p) = 1\}$ .

For instance, when we say that a point  $m$  of  $\mathbb{Z}^2$  belongs to  $I$ , we mean:  $m \in \{p \in \mathbb{Z}^2 \mid I(p) = 1\}$ . The inverse image of  $I$  is  $\bar{I} = 1 - I$ . A point of  $\mathbb{Z}^2$  is also called a pixel. We will use the letters  $p, q, m$  or their coordinates  $(x, y)$  to denote them. We will adopt the usual convention to represent binary images: pixels where the image is equal to 1 will be represented in black, whereas the others will be represented in white.

Let us partition  $\mathbb{Z}^2$  into  $2 \times 2$  blocks. For all  $(x, y)$  in  $\mathbb{Z}^2$ :

$$B(x, y) = \{(2x, 2y), (2x + 1, 2y), \\ (2x, 2y + 1), (2x + 1, 2y + 1)\}. \quad (1)$$

This partition is the base for the construction of the downsampling operator.

**Definition 1 (Binary downsampling operator)** *A binary downsampling operator  $\Delta$  is a function from  $\mathcal{I}$  into  $\mathcal{I}$  such that, for every  $I \in \mathcal{I}$  and  $(x, y) \in \mathbb{Z}^2$ :*

$$(\Delta(I))(x, y) \in \{I(2x, 2y), I(2x + 1, 2y), \\ I(2x, 2y + 1), I(2x + 1, 2y + 1)\} \\ \in I(B(x, y)).$$

Therefore, the main question when defining a binary downsampling operator will be how to choose the value of  $(\Delta(I))(x,y)$  among the set  $I(B(x,y))$ .

A grey level image  $R$  is a function of  $\mathbb{Z}^2$  into  $\{0, \dots, 255\}$ :

$$R: \begin{array}{l} \mathbb{Z}^2 \longrightarrow \{0, \dots, 255\} \\ (x,y) \longmapsto R(x,y) . \end{array}$$

We define  $\text{index\_max}(R, B(x,y))$  as the element of  $B(x,y)$  where  $R$  takes its maximal value. If there were two or more elements of  $B(x,y)$  where  $R$  took its maximal value, then the first of these in video scanning order would be taken.

**Definition 2 (Reference downsampling operator)**

Let  $R$  be a grey level image. The binary downsampling operator  $\Delta_R$  with reference  $R$  is defined as:

$$\Delta_R: \begin{array}{l} \mathcal{I} \longrightarrow \mathcal{I} \\ I \longmapsto \Delta_R(I) = I(\text{index\_max}(R, B(.))) . \end{array}$$

The simplest binary downsampling method, called point sampling, which consists in taking the first pixel of each  $B(x,y)$ , is equivalent to applying a reference downsampling operator with a constant reference image. Needless to say, this method gives very poor results.

The choice of  $R$  is essential to build interesting sampling operators. The objective of this approach is to build  $R$  from  $I$ , in such a way that the value of  $R(x,y)$  corresponds to the importance we want to give to pixel  $(x,y)$  in image  $I$ .

Fig. 1a shows an image to illustrate our purpose. First of all, note that point sampling would produce a completely white image. Methods which favour “black” pixels (pixels belonging to the image) would produce image in Fig. 1b. This would be the case for instance if we used the same initial image as reference image in the binary downsampling operator. The result is considerably better than the result obtained with point sampling, but there has been a topological modification of the image (it will be seen in the next section what is exactly meant by this). Such modifications are often annoying when dealing with binary data. For example, in this case, image (a) would be interpreted as a letter “C”, whereas image (b) would be misunderstood as a letter “O”. We would like to compute a reference image that would give the result shown by image (c) through reference downsampling. To achieve this, pixel  $c$  in the first image should be considered more important than pixels  $a$  and  $b$ , which means that the corresponding value in the reference

image should be larger than the values associated to the other two pixels.

In the next section we will propose a method to build a reference image which takes into account the image topology.

## BUILDING A REFERENCE IMAGE

### DIGITAL TOPOLOGY: A DUAL FRAMEWORK

We recall the main digital topology notions that will be used in the following. For a complete introduction to digital topology, the reader may consult the article by Kong and Rosenfeld (1989).

Let  $\mathcal{N}$  be a neighbourhood relation on  $\mathbb{Z}^2$ , i.e., a binary relation on  $\mathbb{Z}^2$  which is symmetric. When points  $p$  and  $m$  of  $\mathbb{Z}^2$  are in relation through  $\mathcal{N}$ , we say that they are neighbours and we write  $p\mathcal{N}m$ . Moreover, we adopt the following convention: we take  $\mathcal{N}$  such that a point  $p$  is never in relation with itself through  $\mathcal{N}$ . We will denote  $\mathcal{N}(p)$  the set of neighbours of  $p$ . As  $p\mathcal{N}p$  is always false,  $p$  never belongs to  $\mathcal{N}(p)$ .

Two subsets  $A$  and  $B$  of  $\mathbb{Z}^2$  will be said to be  $\mathcal{N}$ -neighbours if they are disjoint and there are two pixels  $m$  and  $p$  respectively belonging to  $A$  and  $B$  such that  $p\mathcal{N}m$ .

Once equipped with a neighbourhood relation, the points of an image can be aggregated into larger structures.

A sequence  $(q_0, \dots, q_K)$  of points of  $\mathbb{Z}^2$ , where  $K$  is a strictly positive integer, is a  $\mathcal{N}$ -path if and only if any two consecutive points of the sequence are  $\mathcal{N}$ -neighbours. The pixels  $q_0$  and  $q_K$  are called the extremities of the path.

Two different points  $m$  and  $p$  belonging to an image  $I$  are said to be  $\mathcal{N}$ -connected in  $I$  if there is a path included in  $I$  whose extremities are  $m$  and  $p$ .

“To be connected in  $I$ ” is an equivalence relation. Its equivalence classes are the  $\mathcal{N}$ -connected components of  $I$ . The number, possibly infinite, of  $\mathcal{N}$ -connected components of a subset of  $\mathbb{Z}^2$  or an image  $I$  will be denoted  $CC_{\mathcal{N}}(I)$ .

We now introduce the notions of interior and isolated point, in a general form adapted to our framework. A point  $p$  of  $\mathbb{Z}^2$  is said to be a  $\mathcal{N}$ -interior point of  $I$  if and only if each of its  $\mathcal{N}$ -neighbours has the same value as itself. A point  $p$  of  $\mathbb{Z}^2$  is said to be a  $\mathcal{N}$ -isolated point of  $I$  if and only if none of its  $\mathcal{N}$ -neighbours have the same value as itself.

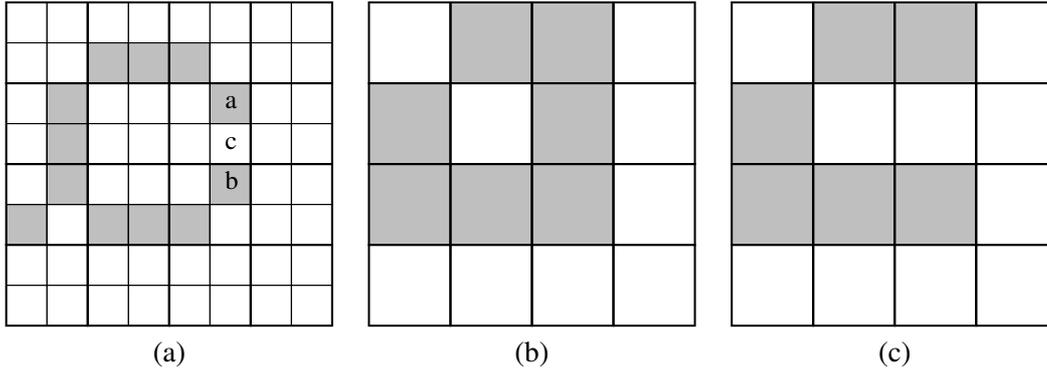


Fig. 1. (a) Example of binary image. (b) Downsampled image when favoring black pixels. (c) Aimed result.

Typical neighbour relations used in image processing are the 4-, 6- and 8- neighbourhoods, respectively denoted  $\mathcal{N}_4$ ,  $\mathcal{N}_8$  and  $\mathcal{N}_6$ . Among these,  $\mathcal{N}_6$  has the best topological properties, as it is the only one that fulfills the digital Jordan curve theorem. But when the image has been digitized following a square grid, 6-neighbourhood causes some unwelcome phenomena. The Khalimsky neighbourhood relation, denoted  $\mathcal{N}_K$ , should also be mentioned. It shows very nice topological characteristics, but it is not translation invariant: if both coordinates of a pixel  $p$  are even or uneven, then  $\mathcal{N}_K(p) = \mathcal{N}_8(p)$ . For all other pixels, we have  $\mathcal{N}_K(p) = \mathcal{N}_4(p)$ .

In order to palliate the defects of 4- and 8- neighbourhoods, neighbourhood relations which depend on the image have been proposed, and widely used. For example, the (8,4)- neighbourhood relation  $\mathcal{N}_{8,4}^I$  is defined as:

$$p \mathcal{N}_{8,4}^I m \Leftrightarrow \begin{cases} p \mathcal{N}_8 m & \text{if } I(p) = 1 \text{ and } I(m) = 1 \\ p \mathcal{N}_4 m & \text{otherwise.} \end{cases} \quad (2)$$

The (4,8)- neighbourhood relation,  $\mathcal{N}_{4,8}^I$ , is defined analogously. We make explicit the dependance of the neighbourhood on the image by putting  $I$  as a superscript on  $\mathcal{N}$ . These image-dependent neighbourhoods fulfill the digital Jordan curve theorem (see Kong and Rosenfeld, 1989, for references to the various demonstrations).

The  $\mathcal{N}$ -homotopy graph of an image can now be introduced. Note that very similar notions are called “adjacency tree” in Kong and Rosenfeld (1989), “homotopy tree” in Serra (1982) and “adjacency graph” in Kong and Roscoe (1985).

**Definition 3 ( $\mathcal{N}$ -homotopy graph)** Let  $I$  be an image and  $\mathcal{N}$  a neighbourhood relation. The  $\mathcal{N}$ -homotopy graph of  $I$  is the non-directed graph whose

vertices are the  $\mathcal{N}$ -connected components of  $I$  and  $\bar{I}$ , and whose edges link  $\mathcal{N}$ -neighbouring connected components. If  $V$  is the set of its vertices, and  $E$  the set of its edges, then the graph will be simply denoted  $(V, E)$ .

When the neighbourhood relation  $\mathcal{N}$  is in fact  $\mathcal{N}_{8,4}^I$  or  $\mathcal{N}_{4,8}^I$ , the homotopy graph is a tree (see Kong and Rosenfeld, 1989, and references within).

Note that this definition is slightly different from the ones given in Kong and Rosenfeld (1989) and Serra (1982). Indeed, no supposition is made about the color of the background. This allows “black” and “white” pixels to play symmetric roles. In fact, if  $\mathcal{N}$  is inversion invariant, then  $I$  and  $\bar{I}$  will have isomorphic graphs, and therefore will be considered homotopic in this framework.

**Definition 4 ( $\mathcal{N}$ -homotopic images)** Let  $\mathcal{N}$  be a neighbourhood relation. Two images are  $\mathcal{N}$ -homotopic if and only if their corresponding  $\mathcal{N}$ -homotopy graphs  $(V_1, E_1)$  and  $(V_2, E_2)$  are isomorphic, i.e., if and only if there is a bijection  $f$  between  $V_1$  and  $V_2$  such that  $(A, A') \in E_1$  if and only if  $(f(A), f(A')) \in E_2$ .

“To be  $\mathcal{N}$ -homotopic” is an equivalence relation. Its equivalence classes are the  $\mathcal{N}$ -homotopy classes of  $\mathcal{I}$ .

The comparison of the homotopy graphs can be used to evaluate the quality of an operator which aims at preserving homotopy. We have manually done this in section “Results” in order to check, for simple configurations, if the proposed adaptive downsampling operator produces a result homotopic to the original image. More generally, a distance between graphs could be used to automatically evaluate the performance from a homotopy point of view.

Let us consider an image  $I$  and a pixel  $p$ . Let  $J$  be the image equal to  $I$  on all pixels of  $\mathbb{Z}^2$  except on  $p$ . When  $p$  belongs to  $I$ , the construction of  $J$  is the essential basic step to compute a thinning operator. For the thinning to be interesting,  $J$  and  $I$  must be homotopic. If this is true, then  $p$  is said to be a *simple point*. More generally, in our dual framework:

**Definition 5 ( $\mathcal{N}$ -simple point)** A point of  $\mathbb{Z}^2$  is  $\mathcal{N}$ -simple with respect to a given image if and only if the inversion of its value does not modify the  $\mathcal{N}$ -homotopy class of the image.

As in image thinning, simple points will play an important role in image downsampling. Indeed, it is important to note that, if the modification of a single simple point does not modify the topology of the considered image, the simultaneous modification of two simple points might do so. This problem, found in image thinning, will appear in our study: the simultaneous disappearance of two simple points through the downsampling procedure might introduce topological modifications.

In 2D, simple points can be simply characterized by the study of their neighbourhood.

## ADAPTIVE CROSSING NUMBERS

The study of the number of connected components of  $\mathcal{N}(p) \cap I$  has lead to several notions, namely the Rutovitz crossing number (Rutovitz, 1966), the Hilditch crossing number (Hilditch, 1969), and the Yokoi connectivity number (Yokoi *et al.*, 1973).

However, these crossing numbers are only defined for pixels belonging to the image. We will now introduce an image inversion invariant crossing number.

In order to define adaptive crossing numbers, we first have to classify the pixels of the image as object pixels or background pixels. Note that we will call object pixels those pixels that are considered important because they belong to the minority in their neighbourhood; they might be black or white. We will then consider an 8-neighbourhood for object pixels, and a 4- neighbourhood for the other pixels.

Consider a pixel  $p$  and a binary image  $I$ . In order to answer the question “does  $p$  belong to the object”, we compute the number  $n^I(p)$  of 8-neighbours of  $p$  where  $I$  takes the same value as on  $p$ . This is given by:

$$n^I(p) = \text{Card}(\{q \in \mathcal{N}_8(p) \mid I(q) = I(p)\}), \quad (3)$$

whose values are included between 0 ( $\mathcal{N}_8$ -isolated point) and 8 ( $\mathcal{N}_8$ -interior point). If this value is equal or

greater than 4, then we will consider  $p$  as a background pixel, otherwise, as an object point.

**Proposition 6** The operator  $n^I$  is invariant with respect to image inversion:

$$\forall I \in \mathcal{I}, \forall p \in \mathbb{Z}^2, n^I(p) = n^{\bar{I}}(p). \quad (4)$$

## Demonstration

$$\begin{aligned} n^I(p) &= \text{Card}(\{q \in \mathcal{N}_8(p) \mid I(q) = I(p)\}) \\ &= \text{Card}(\{q \in \mathcal{N}_8(p) \mid \bar{I}(q) = \bar{I}(p)\}) = n^{\bar{I}}(p) \end{aligned}$$

□

An immediate consequence of this proposition is that the notion of object pixel is inversion invariant.

Let us consider a pixel  $p$ . It is either an object pixel of  $I$ , or a background pixel of  $I$ .

If  $p$  is an object pixel of  $I$  (i.e.,  $n^I(p) < 4$ ), then we consider its 8-neighbours (see Fig. 2a). On some of these neighbours,  $I$  takes a different value from  $p$ ; we call the number of  $\mathcal{N}_4$ -connected components of this subset the *adaptive crossing number* of the object pixel  $p$ .

Similarly, if  $p$  is a background pixel of  $I$  (i.e.,  $n^I(p) \geq 4$ ), then we consider its 4-neighbours (see Fig. 2b). On some of these neighbours,  $I$  takes a different value from  $p$ ; we call the number of  $\mathcal{N}_8$ -connected components of this subset the *adaptive crossing number* of the background pixel  $p$ .

For example, in Fig. 2a, the number of 4-connected components of the set  $\{m \in \mathcal{N}_8(a) \mid I(m) \neq I(a)\}$  is 1, and in Fig. 2b, the number of 8-connected components of the set  $\{m \in \mathcal{N}_4(c) \mid I(m) \neq I(c)\}$  is 2.

More formally:

**Definition 7 (Adaptive crossing number)** The adaptive crossing number of a pixel  $p$  in an image  $I$ , denoted  $X^I(p)$ , is:

$$X^I(p) = \begin{cases} CC_{\mathcal{N}_4}(\{m \in \mathcal{N}_8(p) \mid I(m) \neq I(p)\}), & \text{if } n^I(p) < 4, \\ CC_{\mathcal{N}_8}(\{m \in \mathcal{N}_4(p) \mid I(m) \neq I(p)\}), & \text{if } n^I(p) \geq 4, \end{cases} \quad (5)$$

where  $CC_{\mathcal{N}}(I)$  is the number of  $\mathcal{N}$ -connected components of a subset of  $\mathbb{Z}^2$  or an image  $I$ .

The adaptive crossing number has the property we were looking for:

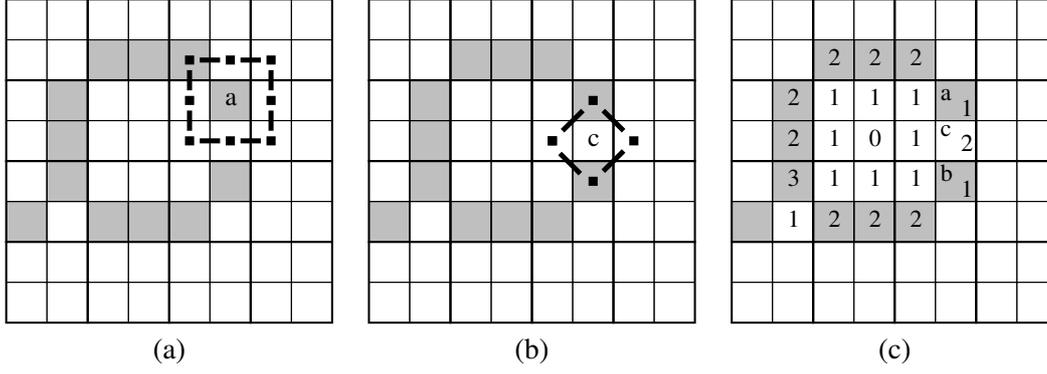


Fig. 2. (a) Pixels around pixel  $a$  considered to compute its adaptive crossing number. Neighbourhood relation on them, used to compute  $X^I(a)$ , indicated by segments. (b) Pixels around pixel  $c$  considered to compute its adaptive crossing number. Neighbourhood relation on them, used to compute  $X^I(c)$ , indicated by segments. (c) Test image with some values of the adaptive crossing number. Notice that the value associated to pixel  $c$  is higher than those given to  $a$  and  $b$ .

**Proposition 8**  $X^I$  is invariant with respect to image inversion:

$$\forall I \in \mathcal{I}, \forall p \in \mathbb{Z}^2, X^I(p) = X^{\bar{I}}(p). \quad (6)$$

**Demonstration** Simply rewrite equation 5 using proposition 6.  $\square$

Note that an adaptive neighbourhood relation could be defined in exactly the same way: 8-neighbourhood would be considered between object pixels, and 4-neighbourhood otherwise. However, this neighbourhood relation does not fulfill the Jordan curve theorem.

As a consequence, the reference image  $R_n^I$  built from  $X^I$  is also invariant with respect to image inversion:

$$R_n^I(p) = \begin{cases} X^I(p) & \text{if } X^I(p) > 0, \\ 5 & \text{otherwise.} \end{cases} \quad (7)$$

The particular case for  $X^I(p) = 0$ , *i.e.*, for isolated points, is necessary if we want these pixels to be preserved. The value 5 is arbitrary; it has to be higher than the other values of  $X^I(p)$ .

Finally, we obtain the following downsampling operator, that we call *adaptive downsampling operator*:

$$\Delta_n(I) = \Delta_{R_n^I}(I), \quad (8)$$

which has the property we were seeking for:

**Theorem 9** The adaptive downsampling operator  $\Delta_n$  is auto-dual:

$$\forall I \in \mathcal{I}, \Delta_n(I) = \overline{\Delta_n(\bar{I})}. \quad (9)$$

**Demonstration** We have  $\Delta_n(\bar{I}) = \Delta_{R_n^{\bar{I}}}(\bar{I})$  and  $R_n^I = R_n^{\bar{I}}$ , therefore, for all  $(x, y)$  in  $\mathbb{Z}^2$ :

$$\begin{aligned} (\Delta_n(\bar{I}))(x, y) &= \bar{I}(\text{index\_max}(R_n^I, B(x, y))) \\ &= 1 - I(\text{index\_max}(R_n^I, B(x, y))) \\ &= \overline{\Delta_{R_n^I}(\bar{I})} \\ &= \overline{\Delta_n(I)}. \end{aligned}$$

$\square$

## RESULTS

First of all, in Fig. 2c we give the values of  $X^I(p)$  for some pixels of the test image. Notice that the value associated to pixel  $c$  is now higher than the values of its neighbours  $a$  and  $b$ . Thanks to this, the resulting downsampled image with the reference image  $R_n$  we have just defined is the one given by Fig. 1c.

Fig. 3 shows some more examples of simple configurations whose topology we would like to preserve.

Fig. 4 gives the result of the adaptive downsampling of the test configurations given in Fig. 3. As it can be seen, in the first case (Fig. 4a), the result is satisfactory. The images before and after downsampling are  $\mathcal{N}$ -homotopic for all usual neighbourhood relations  $\mathcal{N}$  (including 4, 6, and 8 neighbourhoods, as well as (4,8) and (8,4) neighbourhoods). This example also illustrates the

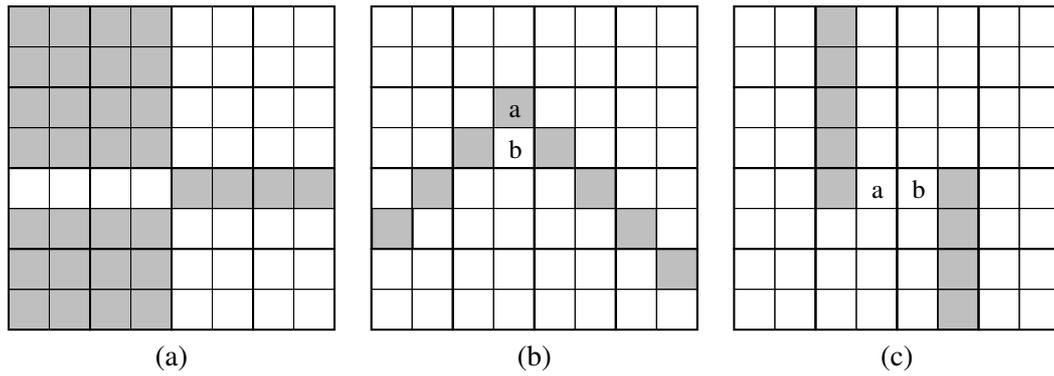


Fig. 3. Test images containing simple configurations.

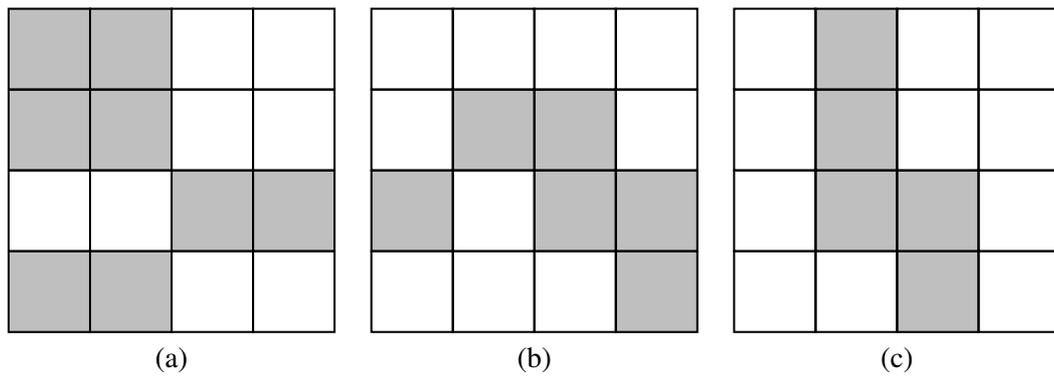


Fig. 4. Adaptive downsampling of test configurations.

auto-dual nature of the adaptive downsampling operator.

The second case (Fig. 4b) is slightly less satisfactory. The images before and after downsampling are for instance (8,4)-homotopic, but not (4,8)-homotopic.

However, the third case (Fig. 4c) has not been conveniently downsampled. It is possible to build a reference image that would have preserved topology, but our method did not allow it. This problem is analogous to the problem of simple points during thinning operations. Indeed, pixels marked *a* and *b* in Fig. 3c have an adaptive crossing number of 1. But, taken together, they are important to preserve the image topology. Their corresponding value in the adaptive reference image should be higher than the corresponding values of the neighbouring black pixels.

Fig. 5 shows a more complex test image, containing geometric structures and text. Its size is  $512 \times 512$ . Fig. 6 shows the result of the application of the adaptive downsampling procedure.

Notice that adaptive downsampling has done a nice work in preserving some important structures. For example, in many cases topological downsampling

has avoided the fusion between letters. However, the proposed downsampling operator has not preserved some geometric details that are also important (look for instance at letters “t”, “j” or “r”). This is not surprising, given that the proposed downsampling operator only aims at preserving topology.

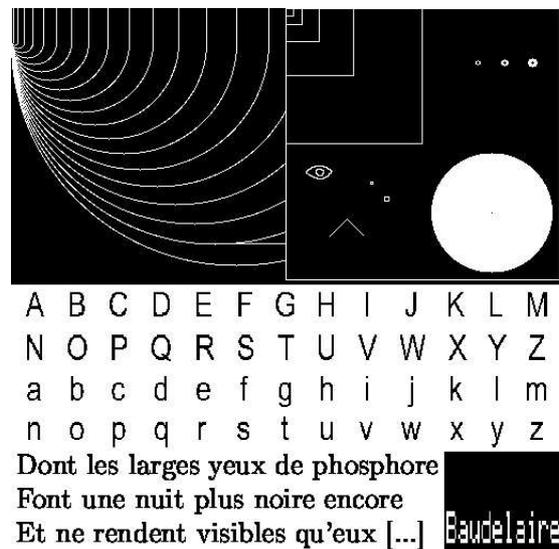


Fig. 5. Test image (512 x 512).

In some other cases (see for example letter “V” in Fig.6b) the topology of some structures has not been preserved. The main reason for this behavior is the lack of space (pixels per letter) in the resulting image. In some other cases, the local analysis does not correctly evaluate the value of some pixels (letters “h” or “y” in Fig.6b).

Finally, it should be noted that given that thin structures tend to be preserved along the downsampling process, their relative size will increase with respect to the larger structures.

In order to compare with state of the art downsampling methods, we have computed a gaussian pyramid from the initial image, see Witkin (1983).

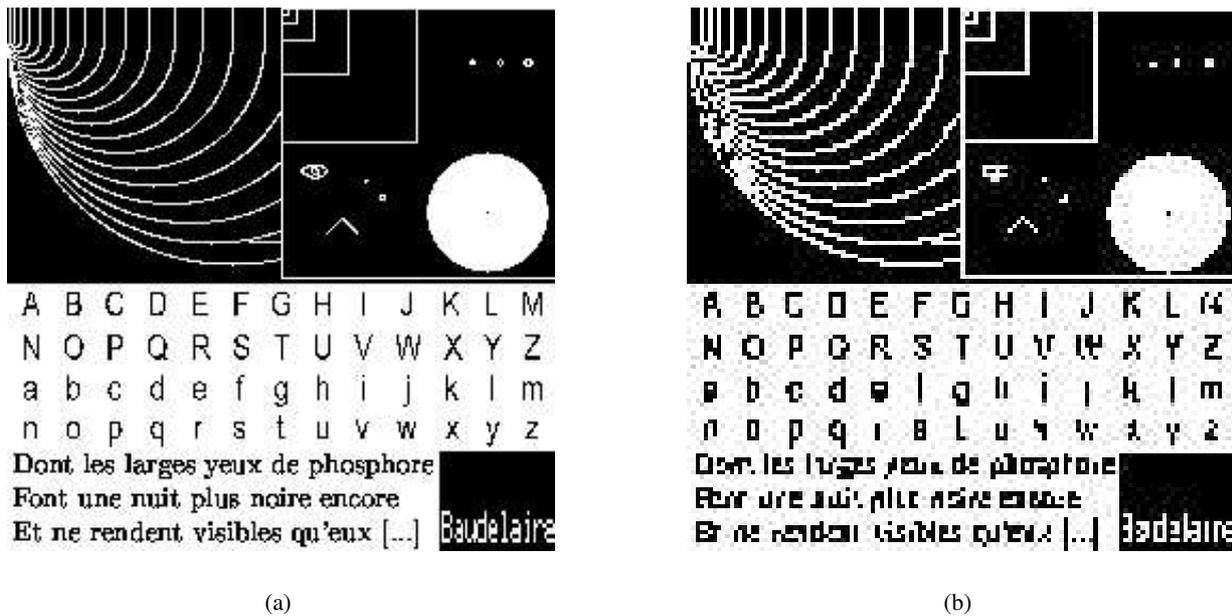


Fig. 6. (a) Adaptive downsampling. (b) Adaptive downsampling, iterated.

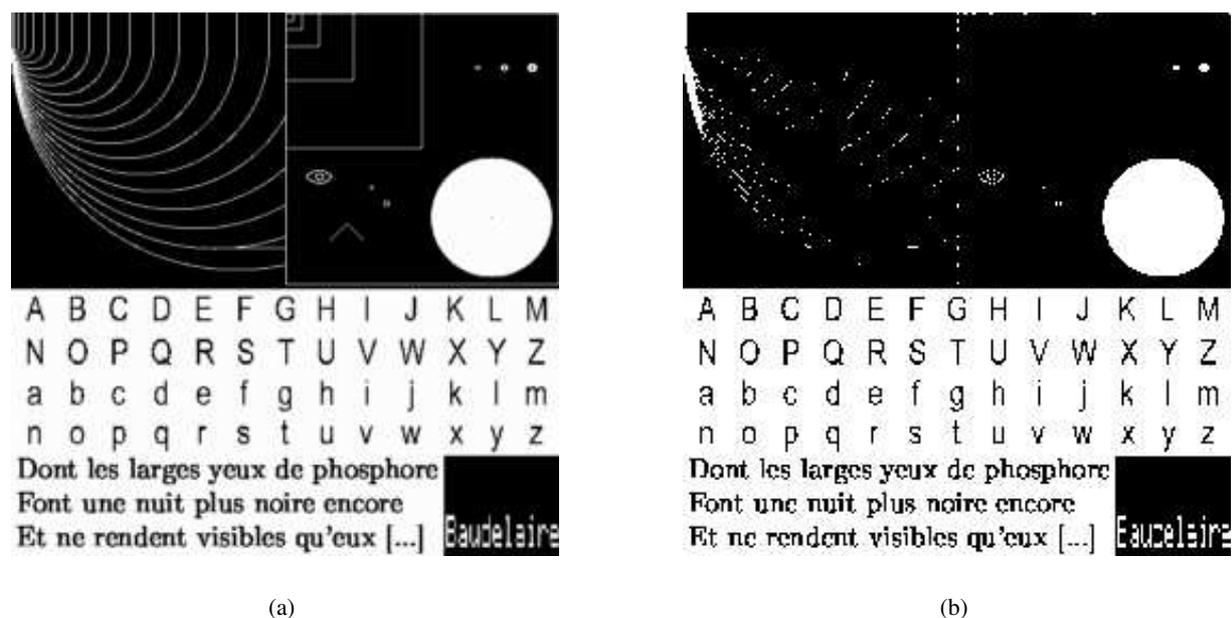


Fig. 7. (a) Linear downsampling, before thresholding. (b) Previous image, thresholded: linear-based binary downsampling.

The original image is first filtered using a  $5 \times 5$  support gaussian, and then point-sampled; the procedure is iterated to obtain the desired number of resolution levels. The resulting images are grey-level (values range between 0 and 255). In order to binarize them, we have used a threshold value of 127.5, which gives a binary downsampling operator which is also auto-dual. Fig. 7 shows the first downsampling step obtained this way. On top (a), the downsampled filtered image is shown, before thresholding. Note that it is a grey-level image, and that it is visually pleasing. However, the threshold (b) produces a much less pleasing result if compared with Fig. 6a. Thin structures have been erased.

## CONCLUSION AND FUTURE DEVELOPMENTS

As far as we know, the binary downsampling method we have presented is the first that uses purely topological criteria in the process.

It does a good job of preserving structures from a topology point of view, but, in some cases, the removal of two neighbouring simple points introduces topological modifications which could have been avoided. Therefore, a more subtle analysis is needed to compute a better reference image. For example, second neighbours could be considered in the analysis.

Moreover, it appears that a topology preservation criterion is not enough to preserve meaningful details. Some geometric information should be added to the reference image, as curvature or information about extremities. The reference downsampling approach allows to combine different sorts of information.

It should be noted that the operations involved in the computation of the reference image are not computationally greedy. The implementation of this method on mobile processors should not be a problem.

The next step in this work will be to extend this downsampling approach to grey level images.

## ACKNOWLEDGMENTS

This work has been financed by the European MEDEA+ Pocket Multimedia project.

## REFERENCES

Borgefors G, Ramella G, Sanniti di Baja G (1996). Multiresolution representation of shape in binary images. In: Miguet S, Montanvert A, Ubéda S, eds. *Discrete Geometry for Computer Imagery*, Proceedings

DCGI'96, vol. 1176 of Lecture Notes in Computer Science. Lyon, France: Springer.

Borgefors G, Ramella G, Sanniti di Baja G (2001). Shape and topology preserving multi-valued image pyramids for multi-resolution skeletonization. *Pattern Recogn Lett* 22:741–51.

Borgefors G, Ramella G, Sanniti di Baja G, Svensson S (1999). On the multiscale representation of 2D and 3D shapes. *Graph Model Im Proc* 61:44–62.

Decenciére E, Bilodeau M (2005). Downsampling of binary images using adaptive crossing numbers. In: Ronse C, Najman L, Decenciére E, eds., *Mathematical morphology: 40 years on*. Proceedings ISMM'2005. Paris, France.

Decenciére E, Marcotegui B, Meyer F (2000). Content dependent image sampling using mathematical morphology. In: Goutsias J, Vincent L, Bloomberg D, eds., *Mathematical Morphology and its applications to signal processing*. Proceedings ISMM'2000. Palo Alto, CA, United States: Kluwer Academic Publishers.

Decenciére E, Marcotegui B, Meyer F (2001). Content-dependent image sampling using mathematical morphology: application to texture mapping. *Signal Process-Image* 16:567–84.

Florêncio D, Schafer R (1994). Critical morphological sampling and its applications to image coding. In: *Mathematical Morphology and its Applications to Image Processing*. Proceedings ISMM'94.

Haralick R, Zhuang X, Lin C, Lee J (1989). The digital morphological sampling theorem. *IEEE Trans Acoust Speech* 37:2067–90.

Heijmans H, Toet A (1991). Morphological sampling. *CVGIP—Image Understanding* 54:384–400.

Hilditch C (1969). Linear skeletons from square cupboards. In: Meltzer B, Michie D, eds. *Machine Intelligence*, Vol. 4. Edinburgh: Edinburgh Univ. Press, 403–20.

Kong T, Roscoe A (1985). A theory of binary digital pictures. *Comput Vision Graph* 32:221–43.

Kong T, Rosenfeld A (1989). Digital topology: introduction and survey. *Comput Vision Graph* 48:357–93.

Rutovitz D (1966). Pattern recognition. *J Royal Statist Soc* 129:504–30.

Serra J (1982). *Image Analysis and Mathematical Morphology—Volume I*. New York: Academic Press.

Witkin AP (1983). Scale-space filtering. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence*.

Yokoi S, Toriwaki J, Fukumura T (1973). Topological properties in digital binary pictures. *Syst Comput Controls* 4:32–40.