Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

A region-based interpolation method for mosaic images

Javier Vidal[1,2], Jose Crespo[1] and Víctor Maojo[1]

[1] *GIB - LIA, Facultad de Informática, Universidad Politécnica de Madrid, Spain*
{jvidal,jcrespo,vmaojo}@infomed.dia.fi.upm.es

[2] *Departamento Ingeniería Informática y Ciencias de la Computación, Facultad de Ingeniería, Universidad de Concepcion, Chile*
jvidal@udec.cl

Abstract Image interpolation is an important operation in some image analysis and processing applications. This paper describes a region-based interpolation method for mosaic images. It is an extension of a previously presented interpolation technique based on morphological operations for binary images. Even though the principles used in this method are similar to those used in the technique for binary images, there are some substantial differences, pointed out in this paper, due the nature of images treated. Some experimental results are provided.

Keywords: mathematical morphology, image processing, image analysis, interpolation, mosaic, median set.

1. Introduction

Image interpolation is necessary in some image processing applications. Methods for image interpolation construct new images from a set of known ones, and they permit to increase the practical resolution of data. However, it should be stated that the interpolated images are obtained from the input images and depend completely on them.

Mathematical morphology [5, 16, 17, 20] techniques have been used to design several interpolation methods [1–4, 7–15, 18, 24] that deal with the shapes of the objects to be interpolated. In this paper, we are especially interested in morphological interpolation for mosaic images [10–12, 24].

The interpolation method for mosaics described in this paper is morphological as well. The principles used are similar to those used in an interpolation technique for binary images presented in [21–23]. These basic principles are two: an inclusion property and the utilization of the median set as interpolator [19]. Despite such similarities, the nature of the treated images make it necessary to treat some additional aspects, which are pointed out in the paper.

The proposed method is region-based. In this work, the input images are 2D mosaic images, i.e., 2D gray-level segmented images that are composed of piecewise-constant regions.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

This paper is organized as follows. Section 2 presents a brief summary of the previously presented interpolation technique. Section 3 presents an extension of this technique in order to treat the so-called border images. Characteristics and technical details of this region-based interpolation method are described in Section 4. Some experimental results are provided in Section 5. Conclusions are commented in Section 6.

2. A brief summary of an interpolation technique for binary images

The technique reported in this paper is based on a previous work of ours applied to binary images [21–23]. It will be briefly described in this section.

2.1 Inclusion property

Our technique is based on an inclusion property that establishes a relationship which we think can greatly facilitate and improve the results of interpolation methods for binary images. It establishes the recursive interpolation of shapes with internal structures (pores and grains).

Formally, if A_i and B_i are two sets of input slice 1, such that $B_i \subset A_i$, and A_j and B_j are two sets of input slice 2, such that $B_j \subset A_j$, and we want to interpolate A_i with A_j, and B_i with B_j, then the following condition should be satisfied:

$$Inter(A_i \setminus B_i, A_j \setminus B_j) = Inter(A_i, A_j) \setminus Inter(B_i, B_j), \tag{1}$$

"*Inter*" corresponds to the interpolation computation. The arguments of *Inter* are the two images to be interpolated and its result is the interpolated one. Figure 1 illustrates inclusion property: Figure 1(b) is equal to Figure 1(e) minus Figure 1(h).

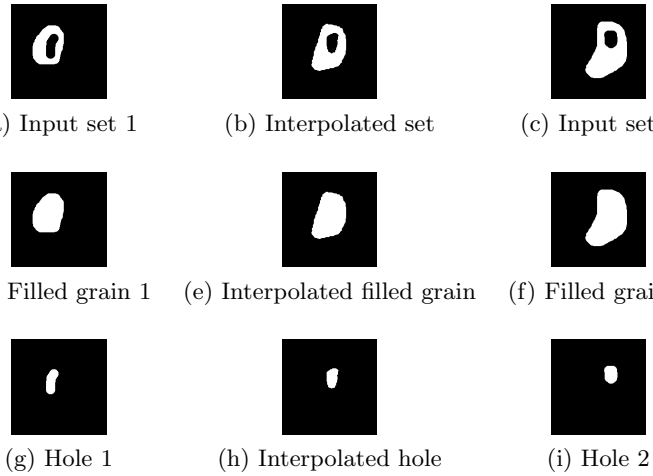
2.2 Binary image interpolation algorithm

In the algorithm for interpolating binary images [21–23], we can distinguish three main sections: (1) separation of outer CCs from each slice, (2) matching of CCs (one from input slice 1 and another from input slice 2), and (3) interpolation of matched CCs. Note that in this technique CC refers to a connected component, and it can denote a grain or a hole.

In the first step, the outer filled CCs of the input slices are identified and separated. The outer filled CCs are the filled CCs surrounded by the background pixels that touch the border of the image. Then, in the matching step, the method establishes correspondences between CCs from the different slices. Those CCs that match will be aligned in order to overlap them and, after that, interpolated using a median set computation.

A detailed description of the algorithm pseudo-code for interpolating binary images can be found in [22].

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45



(a) Input set 1 (b) Interpolated set (c) Input set 2

(d) Filled grain 1 (e) Interpolated filled grain (f) Filled grain 2

(g) Hole 1 (h) Interpolated hole (i) Hole 2

Figure 1. Inclusion property example. Part (d) is the filled grain of part (a). Similarly, part (f) is the filled grain of part (c). Parts (g) and (i) are the holes of, respectively, parts (a) and (c).

3. Treatment of border regions in mosaic images

In the first step of the mosaic interpolation method, regions of connected pixels with the same gray-level are extracted from the pair of input slices and are converted to binary images. Some of the extracted regions touch the border of the images. These regions pose a distinctive problem that must be treated and satisfactorily solved. Next section describes how our technique deals with it. It should pointed out that this issue was not treated in [21–23]).

3.1 Classification of border images

A "binary border image" is an image that has a CC that touches its border. In formal terms, let I and X represent, respectively, a binary image and a connected component (clearly, $X \subseteq I$). If ∂I denotes the set of border

(a) ∂A_1 (b) ∂A_2 (c) ∂A_3 (d) ∂A_4

Figure 2. Border test images.

points of image I (i.e., if the image is $N \times M$, its border pixels are those located at row 0, row $(N-1)$, column 0 and column $(M-1)$), then I is a

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

binary border image if $\partial I \cap X \neq \emptyset$. Figure 4 illustrates several examples of such images[1].

```
BORDERCLASS (X:Slice):integer {
  type = 0;
  IF numberOfCC(X^C) > 1
    type = 8;
  ELSE
    FOR i = 1 TO 4
      IF | ∂A_i ∩ X |=| ∂A_i |
        type = type + 2;
      ELSEIF | ∂A_i ∩ X |≠ 0
        type = type + 1;
  RETURN (type);
}
```

Figure 3. Classification function.

Border CCs are detected and processed by defining four test images, called ∂A_1, ∂A_2, ∂A_3 and ∂A_4. These images are illustrated in Figure 2, and they are composed of a line located at one of the four image borders (up, right, down or left).

The classification of border images is performed by a function described in Figure 3. It consists in intersecting the border and test images. In the classification function in Figure 3, the notation $| Y |$ denotes the cardinal of Y, i.e., the number of pixels of set Y. If X touches all the border ∂A_i, the "type" variable is incremented by 2; if X touches partially the border ∂A_i, "type" is incremented by 1. This is computed for each border. All types of border images are displayed in Figure 4. Note that *Type* 3 and *Type* 5 are not possible. Also, note that *Type* 0 (not illustrated in Figure 4) is a non-border image. We have arbitrarily defined as *Type* 8 the case in which a border CC touches more than one of the border test images ∂A_i and has an intersection with $\cup_i(\partial A_i \cap X)$, $i \in \{1, ..., 4\}$, that is not a connected set.

(a) Type 1 (b) Type 2 (c) Type 4 (d) Type 6 (e) Type 7 (f) Type 8

Figure 4. Border images classification.

[1]Images are surrounded by a gray-frame in order to distinguish the boundaries of the border CCs. Sets and CCs appear in white in this paper.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

3.2 Interpolation of border images

Border CCs detected according to the procedure described in the last section should be interpolated. The interpolation of border CCs is described in the following. Several cases are distinguished. (Note: the region border type is used in case 4.) Let X_1 and X_2 be CCs in two different slices:

1. If X_1 and X_2 are border CCs, such that $X_1 \cap X_2 \neq \emptyset$ then interpolate with X_1 and X_2 in their original location (without alignment). Figure 5 illustrates an example of this situation. Input images are the first and the last one.

Figure 5. Interpolation between border images with non-empty intersection.

2. If just X_1 or X_2 is a border CC and if $X_1 \cap X_2 \neq \emptyset$ then interpolate with X_1 and X_2 in their original location. In Figure 6 is illustrated an example of this situation.

Figure 6. Interpolation between a border image and non-border image with non-empty intersection.

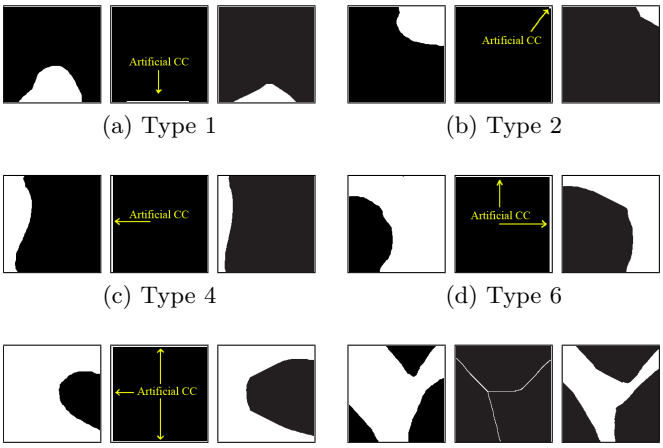
3. If either X_1 or X_2 is a border CC and $X_1 \cap X_2 = \emptyset$ but $\delta_{\lambda_1}(X_1) \cap X_2 \neq \emptyset$ or $X_1 \cap \delta_{\lambda_2}(X_2) \neq \emptyset$ (i.e., these CCs satisfies a proximity test (see later Section 4.2) and are considered a matched pair), then interpolate X_1 and X_2 normally. Figure 7 illustrates an example of this case.

Figure 7. Interpolation between a border image and non border image with empty intersection.

4. In this case, either X_1 or X_2 is the empty set (i.e., it is non-existent). Let us suppose that X_2 is the empty set, so that X_1 vanishes from slide 1 to slide 2. This is dealt in [21–23] with the so-called "artificial" CCs,

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

so that an interpolation is performed between X_1 and an artificial point or line in slide 2. Figure 8 displays several cases of artificial CCs for different types of border CCs. (That is, in this case, the border region classification described in Section 3.1 is employed.) For example, in Figure 8, the Type 2 case uses as artificial CC a point (Figure 8(b)), and the Type 1 case employs a line at the bottom (Figure 8(a)).



(a) Type 1 (b) Type 2

(c) Type 4 (d) Type 6

(e) Type 7 (f) Type 8

Figure 8. Artificial CCs to interpolate isolated border CCs: (a) $X \cap (\bigcup_{k=1}^{4} \partial A_k)$, (b) $X \cap ((\partial A_1 \cap \partial A_2) \cup (\partial A_2 \cap \partial A_3) \cup (\partial A_3 \cap \partial A_4) \cup (\partial A_4 \cap \partial A_1))$, (c), (d) and (e) $\bigcup_{k=1}^{4} X \cap \partial A_k$, if $X \cap \partial A_k = \partial A_k$, (f) $Thinning(A)$. In each case, the left image is X_1, the middle image is the artificial CC, and the right image is the interpolated result.

4. Interpolation of mosaic images

Mosaics are the gray-level images better suited to be treated by a region-based interpolation technique. Morphological interpolation for mosaic images is a recent subject. In the literature, the treatment of shapes in region-based interpolation problems have been frequently involved a conversion to binary images as strategy to treat shapes. In particular, binary morphological skeletons are used in [24] to interpolate shapes of regions. On the other hand, [11, 12] describes a morphological mosaic interpolation based on the distance function calculation. This technique incorporates affine transformations in order to align matched regions (distance functions should be applied to non-empty intersected regions).

The general algorithm of ours region-based interpolation method is displayed in Figure 9. The algorithm is divided in 3 main parts: (1) detection

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

and separation of regions in each slice, (2) matching and interpolation between regions, and (3) final adjustment. Note that these three steps are not exactly identical to the three steps of the binary image interpolation outlined in Section 2; the matching and interpolation stages have been integrated, and the final adjustment step is new (it was not necessary for the binary image case). The next sections describe these parts of the algorithm.

4.1 Region separation

In mosaics, a region corresponds to a set of connected pixels with the same gray-value. In the first step of our algorithm, step (1) in Figure 9, regions are extracted and stored as binary images. The gray-level value for each region is also stored, as well as the hierarchical level that corresponds to the region in a tree structure that is built. A simple mosaic and its regions structure is displayed in Figure 10. Figure 10(a) is the input mosaic, Figures 10(d) and 10(e) are two regions with gray-level value equal to 27, and Figures 10(f) to 10(h) are regions with gray-level values equal to 193, 130 and 93, respectively. Figures 10(b) and 10(c) illustrate the structure of the mosaic and the hierarchical region-based tree of the input image [6].

The first level is composed by all the regions which are directly or indirectly adjacent to the border of the image; the next level includes regions that belong to the internal structures of the regions in the first level in such a way that they are directly or indirectly adjacent to these regions, and so on.

This tree-based information was not necessary in binary images, because the only structure inside a CC that could exist would be a hole or grain inside (and not many regions).

4.2 Matching and interpolation

The objective of the matching step is to establish correspondences between (a) each region in slice 1 and (b) zero or more regions in slice 2. Such a number can be zero if no correspondence occurs. Regions to be matched must have an identical gray-level value as a prerequisite. The matching step corresponds to step (2) in the region-based interpolation algorithm shown in Figure 9.

The criteria used in the matching step are the proximity test and the minimal distances between their MSP[2] points. The proximity test consists in the computation of the *proximity zone*. The proximity zone of X is a dilation of X with a disk-shaped structuring element of radius λ_X. X's

[2]A filled CC, representing a filled region is reduced to a point using the Minimal Skeleton by Pruning [20], i.e., its skeleton is reduced by pruning until a final point is reached. For a filled CC definition see Section 2.2.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

```
INTERPOLATOR (S1, S2:Slice):Slice {
  // (1) Detection and separation of regions in each slice
  For each slice Si
    Store region j from slice Si into vector element RS_i^j
    Compute and store the hierarchical level of RS_i^j
  // (2) Matching and Interpolation (Regions are processed by hierarchical level)
  For each level k of the hierarchical region tree
    For each region i in slice S1
      For each region j in slice S2
        If hierarchical levels of RS_1^i and RS_2^j are k
          // Only regions with the same gray-level are processed
          If graylevel of both RS_1^i and RS_2^j are equal
            If RS_1^i and RS_2^j pass the proximity test
                Establish matching between region i and region j
    // If region i from slice 1 does not match any region in slice 2 or if region j from
    // slice 2 does not match any region in slice 1, the next level is searched
    For each region i in slice S1
      For each region j in slice S2
        If hierarchical level of RS_1^i = k and hierarchical level of RS_2^j = k + 1
        or hierarchical level of RS_1^i = k + 1 and hierarchical level of RS_2^j = k
          If graylevel of both RS_1^i and RS_2^j are equal
            If RS_1^i and RS_2^j pass the proximity test
                Establish matching between region i and region j
  // If there exists multiple matching, only regions with minimal distance
  // between them are kept
  For each region i in slice S1
    If multiple matching is detected for region i
      Choose to match region k from slice S2 with minimal MSP-distance from region i
  For each region j in slice S2
    If multiple matching is detected for region j
      Choose to match region k from slice S1 with minimal MSP-distance from region j
  // Interpolation between matched regions is performed computing median sets.
  For each region i in slice S1
    For each region j in slice S2
      If region RS_1^i match with region RS_2^j
        Compute the interpolated region S_p between RS_1^i and RS_2^j using median set
  // Isolated regions are interpolated with artificial regions (see Figure 8)
  For each region i in slice S1
    If region RS_1^i is isolated
      Choose to match corresponding artificial region for region RS_1^i
      Calculate interpolated region S_p between RS_1^i and its artificial region
  For each region j in slice S2
    If region RS_2^j is isolated
      Choose to match corresponding artificial region for region RS_2^j
      Calculate interpolated region S_p between RS_2^j and its artificial region
  // (3) Final adjustment
  Compute overlapped regions ∪_{i=1,p-1} ∪_{j=i+1,p} (S_i ∩ S_j) and empty regions [∪_{i=1,p} S_i]^C
  Flood overlapped and empty regions with a watershed procedure
}
```

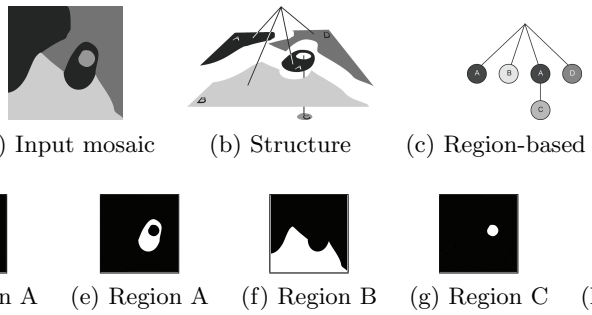
Figure 9. General region-based interpolation algorithm.

radius is computed as $\lambda_X = \min\{\alpha : X \subseteq \delta_\alpha(MSP_X)\}$, where MSP_X is the MSP of X.

If two regions X and Y from different slices match, i.e., $\delta_{\lambda_X}(X) \cap Y \neq \emptyset$ or $X \cap \delta_{\lambda_Y}(Y) \neq \emptyset$, the distance between MSP_X and MSP_Y is stored.

Multiple matching can happen, that is, some regions can match with

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45



(a) Input mosaic (b) Structure (c) Region-based tree

(d) Region A (e) Region A (f) Region B (g) Region C (h) Region D

Figure 10. A mosaic divided in regions.

more than one region of the other slice. Sometimes, this kind of situations produces a merged region. This situation is solved in the following manner: let X be a region from slice 1 that match with n regions $Y_1, Y_2, ..., Y_n$ from slice 2. We must choose Y_k, such that,

$$Y_k = \min_{1 \leq i \leq n} \{Y_i, MSP\_distance(X, Y_i)\},$$

where $MSP\_distance$ represent the Euclidean distance between the MSP points of each input region. The rest of matched regions are rejected.

These criteria and procedures were already used in the interpolation method for binary images [21–23] but, in the case of mosaics, we consider also the hierarchical level of the regions. Particularly, two regions can match if the difference between their hierarchical level is at most one level. This condition is verified before the proximity test in the matching step. The hierarchical level is also used to compute the median sets, i.e., when all the regions at a certain hierarchical level are matched, the median sets between them are computed. After that, correspondences between regions at the next level are analyzed and so on.

Finally, isolated regions are treated. They correspond to regions that do not match and are therefore interpolated with an artificial region whose shape (a point or a line) depends on the position of the region (see examples in Figure 8).

4.3 Final adjustment

This stage is necessary for mosaic image interpolation but not for the binary image case (in fact, this stage was not present in our previous binary image interpolation technique [21–23]). In mosaics, after the previous steps we have a set of interpolated regions $\{R_i\}$ that pose two problems: (a) $\cup_i R_i$ does not necessarily cover the whole image support (i.e., there are empty spaces that do not belong to any R_i); and (b) in general, interpolated regions can overlap (i.e., pixels can belong to more than one interpolated region).

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

These problems are solved at step (3) of the general region-based interpolation algorithm shown in Figure 9. Pixels that belong to empty spaces or to overlapped spaces must be assigned to *one* region. This is performed by using a simple watershed procedure in this last part of the algorithm. Particularly, the image to be flooded is the complement of the image constituted by the interior of $\cup_i R_i$ minus the overlapping regions. Note also that appropriate gray-level values must be used to label the interpolated regions.

5. Experimental results

This section discusses some experimental results of our method. We have used synthetic images that permit to emphasize some relevant aspects of our method.

Figure 11 illustrates a simple case with a human-like mosaic. Input slices are the first and the last one. The body and the two background regions of the images are treated as border regions. Note that the body is Type 8, the big background is Type 6 and the small background is Type 2. The face, eyes and mouth regions are treated as CCs with inclusion relationships using the hierarchical region-based tree.

Figure 11. Example of mosaic interpolation.

The example in Figure 12 shows a situation where there exist an isolated region, e.g., the dark region situated at the bottom right of the image, which vanishes through from the first to the last input image. A border region (the white one at the upper left) that moves to the upper-left corner of the image is also displayed in this sequence.

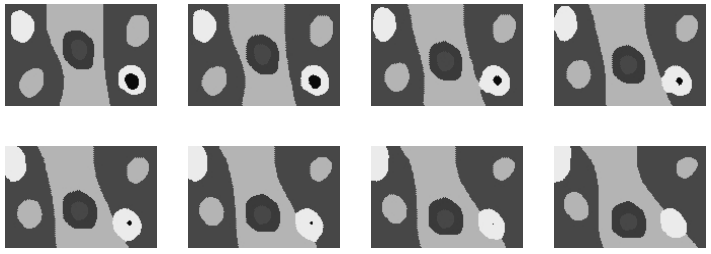


Figure 12. Example of mosaic interpolation.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

6. Conclusions

In this paper, we have presented a region-based interpolation method for mosaic images. It is based on a previous interpolation technique of ours for binary images. As is shown in the paper, even though the basic principles in both techniques are similar, there exist some substantial differences between both cases. For example, the matching and interpolation operations are performed level by level of the hierarchical region-based tree that represents the region structure of the image. The overall interpolation is achieved when all levels have been processed.

Another important difference is that it is necessary a new final adjustment step for mosaic interpolation. After regions have been interpolated, there are generally pixels in the interpolated slice that do not belong to any region. In addition, there can be overlapping between interpolated regions. The last step of our interpolation algorithm for mosaics solve these problems.

Acknowledgments

This work has been supported in part by the University of Concepción (Chile), by the MECESUP programme of the Education Ministry of Chile, by the Artificial Intelligence Laboratory of School of Computer Science of "Universidad Politécnica de Madrid", and by the collaboration programme of Banco Santander-Central Hispano and Universidad Politécnica de Madrid for doctoral studies.

References

[1] S. Beucher, *Interpolations d'ensembles, de partitions et de fonctions*, Technical Report N-18/94/MM, Centre de Morphologie Mathématique, 1994.

[2] ———, *Interpolation of sets, of partitions and functions*, Mathematical morphology ant its applications to image and signal processing, 1998, pp. 119–126.

[3] A. G. Bors, L. Kechagias, and I. Pitas, *Binary morphological shape-based interpolation applied to 3D tooth reconstruction*, IEEE Transaction on Medical Imaging **21** (2002), no. 2.

[4] V. Chatzis and I. Pitas, *Interpolation of 3D binary images based on morphological skeletonizations*, IEEE Transaction on Medical Imaging **19** (2000), no. 7.

[5] E. R. Dougherty and R. A. Lotufo, *Hands-on morphological image processing*, SPIE Press, Bellingham, WA, 2003.

[6] L. Garrido, *Hierarchical region-based processing of images and video sequences: Application to filtering, segmentation and information retrieval*, Ph.D. Thesis, Department of Signal Theory and Communications - Universitat Politècnica de Catalunya, 2002.

[7] J.-F. Guo, Y.-L. Cai, and Y.-P. Wang, *Morphology-based interpolation for 3D medical image reconstruction*, Computarized Medical Imaging and Graphics **19** (1995), no. 3, 267–279.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10 –13, 2007, MCT/INPE, v. 1, p. 201–212.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.16.45

[8] M. Iwanowski and J. Serra, *The morphological - affine object deformation*, International Symposium on Mathematical Morphology (ISMM), palo alto, CA, 2000, pp. 445.

[9] M. Iwanowski, *Application of mathematical morphology to image interpolation*, Ph.D. Thesis, School of Mines of Paris - Warsaw University of Technology, 2000.

[10] _____, *Generalized morphological mosaic interpolation and its application to computer-aided animations*, Computer analysis of images and patterns, 2001, pp. 494–501. Proceedings of 9th International Conference CAIP 2001, Sept. 5-7, 2001, Warsaw, Poland.

[11] _____, *Image morphing based on morphological interpolation combined with linear filtering*, International journal of WSCG, 2002, pp. 233–239. Proceedings of the 10th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Pilzen, Czech Republic.

[12] _____, *Universal morphological interpolator* (I. S. P. Society, ed.), Vol. II, 2005. Proceedings of International Conference on Image Processing ICIPŠ05, Genova, Italy.

[13] T.-Y. Lee and W.-H. Wang, *Morphology-based three-dimensional interpolation*, IEEE Transactions on Medical Imaging **19** (2000), no. 7, 711–721.

[14] F. Meyer, *Interpolations*, Technical Report N-16/94/MM, Centre de Morphologie Mathématique, 1994.

[15] _____, *A morphological interpolation method for mosaic images*, Mathematical morphology and its applications to image and signal processing, 1996.

[16] J. Serra, *Mathematical morphology. Volume I*, London: Academic Press, 1982.

[17] _____(ed.), *Mathematical morphology. Volume II: Theoretical advances*, London: Academic Press, 1988.

[18] _____, *Interpolations et distances of Hausdorff*, Technical Report N-15/94/MM, Centre de Morphologie Mathématique, 1994.

[19] P. Soille, *Spatial distributions from contour lines: an efficient methodology based on distance transformations*, Journal of Visual Communication and Image Representation **2** (1991), no. 2, 138–150.

[20] _____, *Morphological image analysis: Principles and applications*, 2nd ed., Springer-Verlag, 2003.

[21] J. Vidal, J. Crespo, and V. Maojo, *Inclusion relationships and homotopy issues in shape interpolation for binary images*, Lecture Notes in Computer Science, LNCS 3429, 2005, pp. 206–215.

[22] _____, *Recursive interpolation technique for binary images based on morphological median sets*, Computational Imaging and Vision Journal, 2005, pp. 53–62.

[23] _____, *A shape interpolation technique based on inclusion relationships and median sets*, Accepted in "Image and Vision Computing Journal" (2007).

[24] D. N. Vizireanu, S. Halunga, and O. Fratu, *A grayscale image interpolation method using new morphological skeleton*, Proceeding of 6th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services, 2003, pp. 519–521.