

Inclusion Relationships and Homotopy Issues in Shape Interpolation for Binary Images

Javier Vidal^{1,2}, Jose Crespo¹, and Victor Maojo¹

¹ Artificial Intelligence Laboratory, Facultad de Informática,
Universidad Politécnica de Madrid,
28660 Boadilla del Monte (Madrid), Spain
jvidal@infomed.dia.fi.upm.es
{jcrespo, vmajo}@fi.upm.es

² Computer Science Department, Universidad de Concepción, Chile
jvidal@udec.cl
<http://www.udec.cl/~jvidal>

Abstract. Some image processing and analysis applications require performing image interpolation. This paper focuses on interpolation techniques that treat the shapes and the structures of binary images. A summary of some interpolation methods is presented, and their behavior concerning inclusion relationships and homotopy issues is studied. Then, this work discusses an inclusion relationship property that is used in a technique of ours based on median sets that has been recently proposed. The paper shows that such a property can improve shape interpolation results in a relatively easy manner. Several experimental results are provided.

Keywords: image processing, interpolation, shape interpolation, mathematical morphology, median set.

1 Introduction

Image processing applications normally deal with discrete data organized in slices. In some situations, it is useful, or even required, to perform some type of image interpolation [1]. The objective of image interpolation methods is generally to compute new interpolated slices between those originating from the original data.

Until recently, almost all image interpolation methods have used traditional numerical analysis techniques and linear signal processing approaches. In the last decade, several methods that deal directly with image shapes have been proposed [2, 3, 4, 5, 6, 7, 8, 9], in particular mathematical morphology [10, 11, 12, 13] related approaches.

This paper focuses on the behavior of image interpolation methods concerning inclusion relationships and homotopy issues in binary images. First, we study the behavior of some methods in several situations. We will be concerned, especially, in what happens when nested grains and pores are present in an image.

Then, we discuss a simple inclusion property satisfied by a technique of ours that has been recently proposed [14]. We will see that this property can improve the experimental results obtained in situations where there exists an inclusion of structures or shapes. There are several aspects in this technique of relative complexity (such as, for example, how connected components are matched), but we will focus in how the application of the inclusion property can be beneficial in these situations.

This paper is organized as follows. Section 2 provides a summary of some interpolation methods. In Section 3, we study the behavior of the methods presented in Section 2 in some experimental situation; afterwards, we discuss the inclusion property of our technique and its application to several experimental cases. Conclusions are commented in Section 4.

2 A Summary of Some Shape Interpolation Methods

This section presents a summary of the interpolation methods focusing mainly on morphological ones, which adequately consider images shapes.

1. *The methods based on the Hausdorff distance.* The usage of Hausdorff distance for image interpolation was introduced in [9, 15]. A general definition of the Hausdorff distance between two binary sets X and Y is:

$$\rho(X, Y) = \inf\{\lambda : X \subseteq \delta_\lambda(Y); Y \subseteq \delta_\lambda(X)\} \tag{1}$$

where δ_λ denotes the dilation by a compact disk of radius λ centered at the origin.

The most basic Hausdorff geodesic between X and Y is obtained by:

$$Z_\alpha(X, Y) = \delta_{\alpha\rho}(X) \cap \delta_{(1-\alpha)\rho}(Y), \alpha \in [0, 1] \tag{2}$$

where the coefficient α is incremented from 0 to 1 in order to generate the different sets Z .

A possible problem of (2) is that the interpolated set Z_α depends on the distance between the input sets and, in general, its size is bigger than X and Y . Other related methods based on Hausdorff distance intend to correct this issue. For example, an additional Hausdorff geodesic corresponding to the cross-dilation $Z'_\alpha = \delta_{\alpha Y}((1-\alpha)X)$ can be used to intersect (2). Another variant, the reduced Hausdorff distance method, defines X_a and Y_b as the translated initial sets X and Y using vectors a and b , respectively. This way, the initial sets are “aligned” before applying (2). Finally, the incorporation of a mask G (to be intersected with (2)) was proposed in [6, 16]. The mask G is obtained as $G = \varepsilon_\lambda(\text{CH}(\delta_\lambda(X \cup Y)))$, where CH computes the convex hull of a set [12].

2. *Median set-based interpolation.* The median set [2, 4] is related to the morphological notion of skeleton by influence zone (SKIZ).

Let us consider two sets X and Y , such that $X \cap Y \neq \phi$. The median set can be defined as:

$$M(X, Y) = \bigcup_{\lambda \geq 0} \{\delta_\lambda(X \cap Y) \cap \varepsilon_\lambda(X \cup Y)\} \tag{3}$$

where δ_λ and ε_λ represents the dilation and erosion with a disk of radius λ . Initially this method was restricted to intersected sets. Subsequently, the notion of median set to non-intersected sets using affine transformations (translation, rotation and scaling) is described in [6, 17]. The interpolation sequence is obtained computing iteratively (3).

3. *Interpolation function.* The method above can also be implemented using two relative distances to generate the interpolated set [8]. This technique is also applied to intersected sets.

Let X and Y be the input sets, such that $X \cap Y \neq \phi$. The interpolation functions corresponds to the distance from $X \cap Y$ in X ($\text{int}_{X \cap Y}^X$) and to the distance from $X \cap Y$ in Y ($\text{int}_{X \cap Y}^Y$):

$$\text{int}_{X \cap Y}^X(x) = \begin{cases} 0; & x \in X^c \\ d(x); & x \in \frac{X}{(X \cap Y)} \\ 1; & x \in (X \cap Y) \end{cases} \quad \text{int}_{X \cap Y}^Y(x) = \begin{cases} 0; & x \in Y^c \\ d(x); & x \in \frac{Y}{(X \cap Y)} \\ 1; & x \in (X \cap Y) \end{cases} \tag{4}$$

where d is the relative distance from x to $X \cap Y$ and from x to X^c (in the left case) or to Y^c (in the right case). The value of d is between 0 and 1.

The interpolated set Z at distance α from X and $(1 - \alpha)$ from Y is the union of two sets:

$$Z_\alpha = \{x : \text{int}_{X \cap Y}^X(x) \leq \alpha\} \cup \{y : \text{int}_{X \cap Y}^Y(y) \leq (1 - \alpha)\}; \quad \alpha \in [0, 1] \tag{5}$$

The interpolation function can be applied to non-intersected sets using affine transformations [6].

This summary is not exhaustive, and other methods were also studied during the early stages of our work. For example we can mention the methods described in [5, 18, 19, 20, 7].

3 Inclusion Relationships and Homotopy Issues

We think that the consideration of inclusion relationships and homotopy aspects is important to adequately interpolate image shapes.

In this section, we will first study the behavior of the methods presented in Section 2 concerning homotopy issues with binary images. Then, we will see that many of these situations can be better treated with a technique of ours that implements a simple inclusion property.

3.1 Behavior of Interpolation Methods

In the following, we will discuss some results obtained with the interpolation methods described in Sec. 2.

Figure 1 displays and interpolation computed using the Hausdorff distance method with convex mask (computed as described in Sec.2). The two input slices, Fig. 1(a) and Fig. 1(k), have the same homotopy. The intermediate slices, with α -values from 0.1 to 0.9, are the interpolated ones. We can clearly see that the homotopy changes in the sequence of images. The hole disappears in Figs. 1(c)-(i). We can also mention that the interpolated shapes do not sometimes correspond satisfactorily to the shapes observed in the input slices.

We use in this paper homotopy trees [12] to easily visualize the homotopy of an image. Two binary images are homotopic if their respective homotopy trees are the same¹.

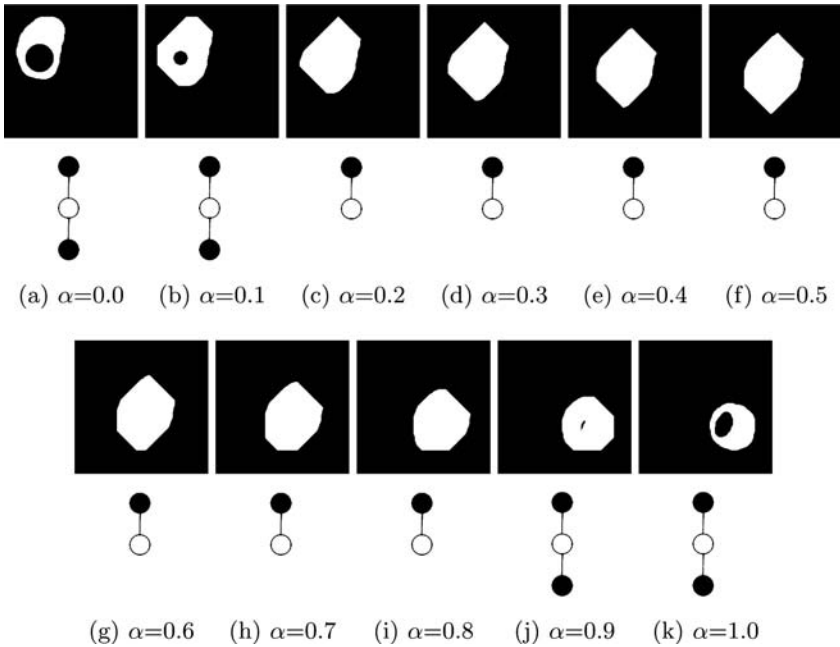


Fig. 1. Sequence of interpolated slices obtained using interpolation based on Hausdorff distance with convex hull and their homotopy trees

A result computed using the median set method (with affine transformation) is shown in Fig. 2. The input slices are displayed in Fig. 2(a) and in Fig. 2(j). In

¹ Nodes corresponding to the background and holes appear in black, and nodes corresponding to grains are displayed in white. The upper nodes of the trees correspond to the outer background.

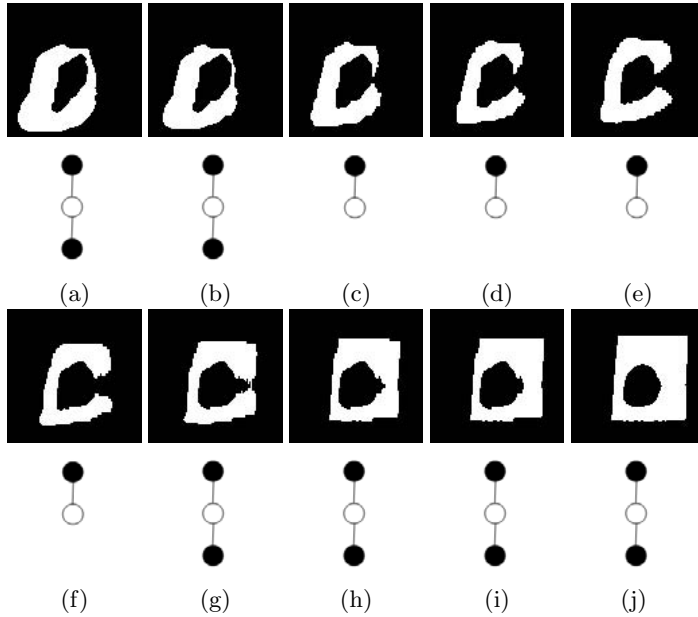


Fig. 2. Sequence of interpolated slices generated by median set-based interpolation method and their homotopy trees

this example, the homotopy of the input slices is also the same. However, we can see clearly that the homotopy of the interpolated slices of Figs. 2(c)-(f) differs from that of the input slices: the ring-shaped connected component “breaks”, and it does not surround a hole inside.

Figure 3 shows a result computed using the interpolation function method. In this case the homotopy changes several times in the interpolated image sequence, as can be clearly observed in the associated homotopy trees.

3.2 The Inclusion Property and Experimental Results

We have proposed [14] an interpolation technique that satisfies the inclusion property that is stated next. It is not our intention to treat here the details of the algorithm, but to emphasize that this property can improve shape interpolation results in a relatively easy manner.

Inclusion Property: Suppose A_1 and B_1 are two sets of the input slice 1, such as $A_1 \subset B_1$, and suppose C_2 and D_2 are two sets of input slice 2, such as $C_2 \subset D_2$. If there is a correspondence between these two pairs (i.e., we want to interpolate A_1 with C_2 , and B_1 with D_2), then the following condition should be satisfied:

$$\text{Interpolate}(A_1 \setminus B_1, C_2 \setminus D_2) = \text{Interpolate}(A_1, C_2) \setminus \text{Interpolate}(B_1, D_2) \quad (6)$$

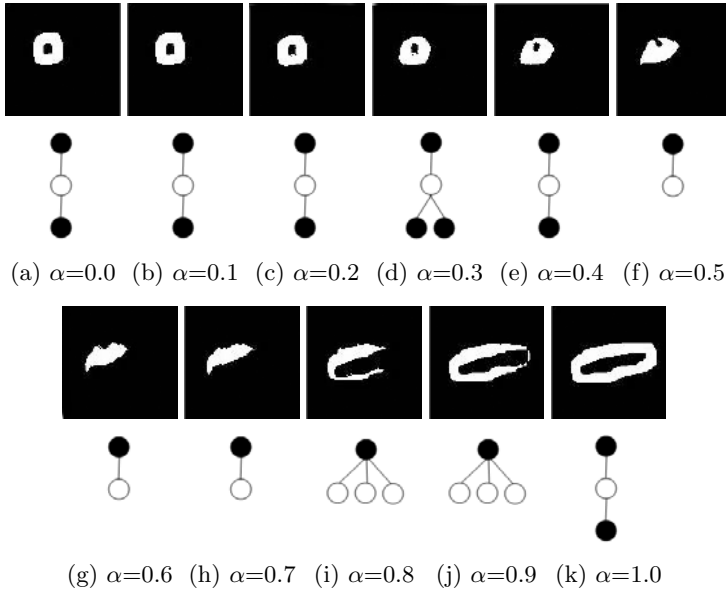


Fig. 3. Sequence of interpolated slices generated by the interpolation function method and their homotopy trees. (Note that there are two pores in image (d) and three grains in images (i) and (j).)

where “Interpolate” denotes our interpolation technique, and “\” symbolizes the set difference. The arguments of “Interpolate” are the two input images, and the result will be an interpolated slice.

The inclusion property is illustrated in Fig. 4. If (6) is satisfied, then image in Fig. 4(h) is equal to the image in Fig. 4(b) minus the image in Fig. 4(d).

Our interpolation technique is based on median sets [4]. Moreover, this technique processes recursively the connected components (CCs) of the slices applying (6) to the inner holes and inner grains of the images. The pseudo-code of the algorithm that implements this technique is shown in the Appendix, and a complete description can be found in [14]. The recursive application of (6) is performed by the last sentence of the while loop in the pseudo-code.

We will discuss next the performance of our technique over the images previously used in Sec. 3.1, and with a more complex case as well.

Figure 5 and Fig. 6 display the application of our technique to the cases visualized in Fig. 2 and Fig. 3, respectively. In both cases, the input slices contain a ring-like CC (a grain with an hole inside). Our technique first interpolates the filled grains. We suppose that the correspondence between them have been established using a matching procedure, and that the CCs are aligned (see [14] for details).

Then, in the next recursive iteration, pores are independently interpolated. The overall interpolation result is the result of interpolated grain filled minus the result of the interpolated pore. This is clearly what (6) establishes. We can

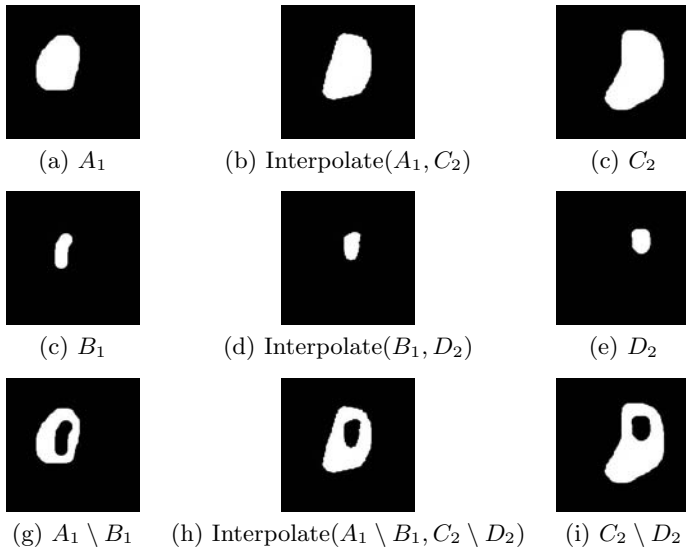


Fig. 4. An example of the inclusion property

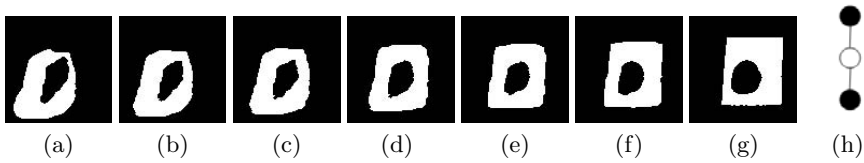


Fig. 5. Interpolated slices using our technique (compare with Fig. 2)

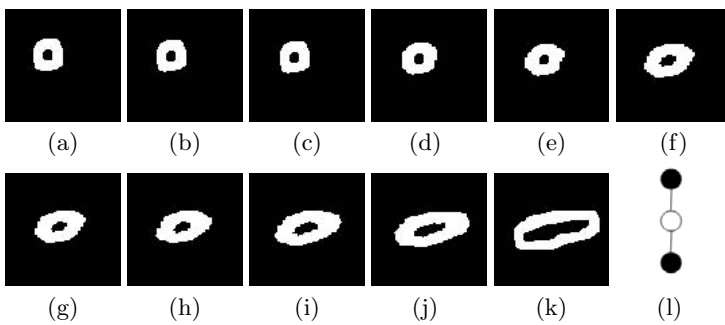


Fig. 6. Interpolated slices using our technique (compare with Fig. 3)

observe that all intermediate interpolated slices are a grain with an hole inside (the same homotopy as that of the input slices), and that these results compare favorably with those obtained in Fig. 2 and in Fig. 3.

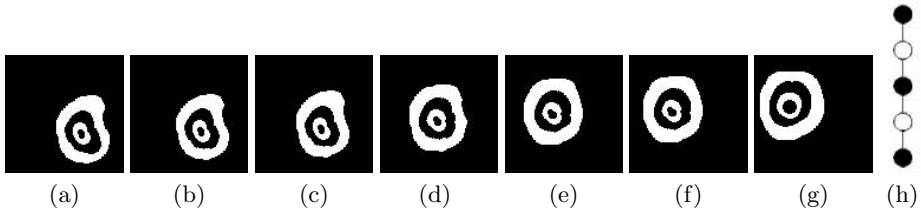


Fig. 7. Interpolation of nested CCs

Finally, we will present the application of expression (6) of the inclusion property to a more complex situation. In Fig. 7, we deal with nested CCs. In this case, our technique applies recursively (6) to the inner structures. Again, we can say our technique obtains satisfactory results.

4 Conclusions

In this paper we have studied homotopy related issues that arise in binary image interpolation. We have commented the behavior of some interpolation methods in some situations, and we have observed that problems can sometimes arise. We have discussed a technique based on median set that applies a relatively simple inclusion property, which can significantly improve the experimental results. Nevertheless, all homotopy issues are not completely solved. A complete treatment of them would require to consider how the connected components are matched and aligned, and the relative position of the nested shapes.

Acknowledgements. This work has been supported in part by the University of Concepción (Chile), by the MECESUP programme of Chilean Education Ministry, and by the Artificial Intelligence Laboratory, School of Computer Science at “Universidad Politécnica de Madrid”.

References

- [1] Meijering, E.: A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE* **90** (2002) 319–342
- [2] Soille, P.: Spatial distributions from contour lines: an efficient methodology based on distance transformations. *Journal of Visual Communication and Image Representation* **2** (1991) 138–150
- [3] Herman, G.T., Zheng, J., Bucholtz, C.A.: Shape-based interpolation. *IEEE Computer Graphics and Application* **12** (1992) 69–79
- [4] Beucher, S.: Interpolations d’ensembles, de partitions et de fonctions. Technical Report N-18/94/MM, Centre de Morphologie Mathématique (1994)
- [5] Guo, J.F., Cai, Y.L., Wang, Y.P.: Morphology-based interpolation for 3D medical image reconstruction. *Computerized Medical Imaging and Graphics* **19** (1995) 267–279

- [6] Iwanowski, M.: Application of Mathematical Morphology to Image Interpolation. PhD thesis, School of Mines of Paris - Warsaw University of Technology (2000)
- [7] Lee, T.Y., Wang, W.H.: Morphology-based three-dimensional interpolation. *IEEE Transactions on Medical Imaging* **19** (2000) 711–721
- [8] Meyer, F.: Interpolations. Technical Report N-16/94/MM, Centre de Morphologie Mathématique (1994)
- [9] Serra, J.: Interpolations et distances of Hausdorff. Technical Report N-15/94/MM, Centre de Morphologie Mathématique (1994)
- [10] Serra, J.: *Mathematical Morphology*. Volume I. London: Academic Press (1982)
- [11] Serra, J., ed.: *Mathematical Morphology*. Volume II: Theoretical advances. London: Academic Press (1988)
- [12] Soille, P.: *Morphological Image Analysis: Principles and Applications*. 2nd edn. Springer-Verlag (2003)
- [13] Dougherty, E.R., Lotufo, R.A.: *Hands-on Morphological Image Processing*. SPIE Press, Bellingham, WA (2003)
- [14] Vidal, J., Crespo, J., Maojo, V.: Recursive interpolation technique for binary images based on morphological median sets. In: *accepted in International Symposium on Mathematical Morphology (ISMM'05)*, Paris, France. (2005)
- [15] Serra, J.: Hausdorff distances and interpolations. In Heijmans, H.J., Roerdink, J.B., eds.: *Mathematical Morphology and its Applications to Images and Signal Processing*, Dordrecht, The Netherlands. Kluwer Academics Publishers (1998)
- [16] Iwanowski, M.: Morphological binary interpolation with convex mask. In: *Proceedings International Conference on Computer Vision and Graphics*, Zakopane, Poland. (2002)
- [17] Iwanowski, M., Serra, J.: The morphological-affine object deformation. In John Goutsias, L.V., Bloomberg, D.S., eds.: *International Symposium on Mathematical Morphology*, Palo Alto, CA. Kluwer Academics Publishers (2000) 445
- [18] Migeon, B., Charreyron, R., Deforge, P., Marché, P.: Improvement of morphology-based interpolation. In: *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. (1998) 585–587
- [19] Chatzis, V., Pitas, I.: Interpolation of 3d binary images based on morphological skeletonizations. In: *Proceedings IEEE International Conference on Multimedia Computing Systems*, Florence, Italy. Volume II. (1999) 939–943
- [20] Chatzis, V., Pitas, I.: Interpolation of 3D binary images based on morphological skeletonizations. *IEEE Transaction on Medical Imaging* **19** (2000) 699–710

Appendix: Interpolation Algorithm Pseudo-Code

Our interpolation technique [14] combines the use of median sets with the recursive application of (6), i.e., the inclusion property.

The algorithm consists of three main steps: (A) the separation of outer filled CCs within the input slices, (B) the matching between the fill CCs of input slices, and (C) the interpolation step that includes the recursive application of (6). The CCs matching step, step (B), determines the correspondence between the CCs (grains or pores) of the two input slices at each recursive iteration.

INTERPOLATOR(*Current_S1* : *Slice*, *Current_S2* : *Slice*) : *Slice*

```

≡ [
  // (A) Separation of outer CCs
  OS1 = Extract_Outer_CC(Current_S1);
  OS2 = Extract_Outer_CC(Current_S2);
  // (B) Matching
  P = MATCHING(OS1, OS2);
  // P is a vector of pairs of matched CCs and their MSPs
  S = ∅; // init result binary image (set notation)
  // (C) Interpolation
  while P ≠ ∅ do
    (FS1, MSP1, FS2, MSP2) = Extract_Pair_CCs(P);
    // (FS1, FS2) is a pair of matched filled CCs
    // Computation of median set MS
    MS = MSF(FS1, MSP1, FS2, MSP2);
    // MSF : median set function using MSPs
    // Holes are stored as CCs
    HS1 = (FS1 ∩ Current_S1)C \ [FS1]C;
    HS2 = (FS2 ∩ Current_S2)C \ [FS2]C;
    // Result image is updated; inner structures
    // (holes and grains) are treated recursively
    if ((HS1 = ∅) ∧ (HS2 = ∅))
      then S = S ∪ MS;
      else S = S ∪ [MS \ INTERPOLATOR(HS1, HS2)]; // Recursive call
    fi
  od
  Return(S);
]

```

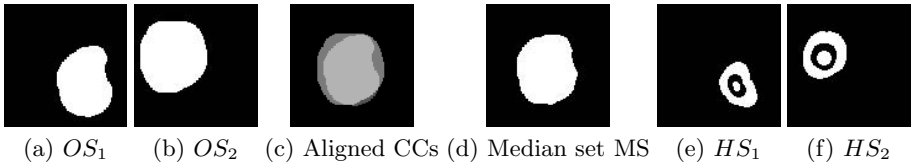


Fig. 8. Algorithm execution example

In the following, we will briefly comment the execution of the algorithm applied to the input slices of Fig. 7. Input slices are those in Fig. 7(a) and Fig. 7(g). In the first iteration, step (A) extracts the outer filled grains (Fig. 8(a) and Fig. 8(b)) of the input slices. The matching step, step (B), matches these extracted CCs (Fig. 8(c) displays the CCs superimposed and aligned as part of the matching process; see details in [14]). Step (C), the main interpolation step, first computes the median set of the aligned filled CCs (Fig. 8(d)). Then, the inner grains and pores are extracted and complemented (Fig. 8(e) and Fig. 8(f)), and the recursive call is performed. Finally, the interpolated slice after all recursive iterations (in this case, 4 iterations) can be found in Fig. 7(d).