Morphological Interpolation for Image Coding

Josep R. Casas

Signal Theory and Communications Department, Universitat Politècnica de Catalunya
Campus Nord UPC, edifici D5, Gran Capità s/n
08034 Barcelona, Spain
E-Mail: josep@gps.tsc.upc.es

Keywords

Morphological interpolation, perceptual image models, feature-based image coding, mathematical morphology

Abstract

The aim of this paper is to present a new interpolation technique intended for spatial interpolation from sparse data sets. The proposed implementation, based on non-linear morphological operators, overperforms linear interpolation techniques based on diffusion processes performing iterative space-variant filtering on the initial image.

The application of morphological interpolation is illustrated for sketch-based image coding. We put forward a perceptually motivated two-component image model that strongly relies on morphological operators. The watershed is used to detect strong edge features in the first component of the model. The smooth areas of the image are recovered from the extracted edge information by morphological interpolation. The residual component, containing fine textures, is separately coded by a subband coding scheme.

I. Introduction

Interpolative coding techniques 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 usually approximated by continuous functions with some permissible error at the interpolated positions. The subset of transmitted pixels, called the *initial set* in the following, may be either a regular subsampling grid or any arbitrary set of points. In the latter case, both the amplitude values and positions of the pixels of the initial set should be coded and transmitted.

The target of image coding is to represent an image or an image sequence with as few bits as possible. As we seek lower bit rates in the digital representations of image data, it is imperative to design the compression algorithm both to reduce redundancy in the input signal and to remove irrelevant information from a perceptual point of view. Perceptual-based image coding models [5] put special emphasis on this second operation. Current standards for image compression [14] exploit some aspects of visual perception –for instance, in the design of quantization tables for DCT coefficients– but it is generally accepted that only the study of image models strongly related to the Human Visual System [6] will lead to the highest compression ratios needed for very low bit-rate applications. The so-called Second Generation models permit a graceful degradation of the perceived quality of reconstructed images at low bit-rates, without the unnatural artifacts (blockiness, ringing and blurring) of waveform coding techniques.

A. Interpolation and sketch-based image coding

A number of different approaches using interpolation techniques have been reported in the literature for 'perceptually motivated' coding applications [1], [4], [11]. The underlying image model is based on the concept of the "raw primal sketch" [8]. The image is assumed to be made mainly of areas of constant or smoothly changing intensity separated by discontinuities represented by strong edges. The coded information, also known as *sketch data*, consists of the geometric structure of the discontinuities and the amplitudes at the edge pixels. In very low bit-rate applications, the decoder has to reconstruct the smooth areas in between using just this information. This can be posed as a scattered data interpolation problem from arbitrary initial sets (the sketch data) under certain smoothness constraints. For higher bit-rates, the residual texture information is separately coded by means of a waveform coding technique; for instance, pyramidal or transform coding.

The performance of such perceptual model has been thoroughly investigated [11], proving its utility for most coding applications and showing subjective improvements over DCT-based methods, as JPEG, at low bit-rates. However, one of the important drawbacks is the large computation time spent in the interpolation process. The techniques proposed to solve the problem of interpolation from sparse data are based on the solution of an energy minimization (variational) problem, governed by the heat or diffusion equation. The practical implementations make use of iterative space-variant filtering operations that converge rather slowly to the final interpolated image.

This work was supported by the European RACE/Morpheco project 2053

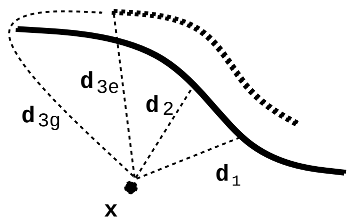


Fig. 1. Geodesic distance measure for the interpolation of pixel x

The aim of this paper is to present a morphological technique intended to perform spatial interpolation from any set of initial pixels. The algorithm, described in section II, is based on morphological operators: namely geodesic dilation and the morphological Laplacian, resulting in a highly efficient process compared to those that perform linear filtering on the initial image. Comparative figures of computation time will be given to assert the efficiency of the process. Its application to sketch-based coding is illustrated in section III, where a two-component model is proposed for perceptual image coding. Finally, some coding results and the conclusions are presented in the last section of the paper.

II. Morphological interpolation technique

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 (the initial set). Such surface is constrained to be maximally smooth between the known pixels in the sense that pixel to pixel variations in the interpolated area should be minimized.

A suitable strategy for spatial 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 [13]. The nearest pixels have stronger influence than the distant ones, and the interpolated amplitudes change slowly in the areas in between.

A. Geodesic distance weighting

The distance measure is taken as the *geodesic distance* within the set of unknown pixels, that is, the length of the shortest path joining two points which is completely included within the set of unknown pixels. The use of the geodesic distance allows the preservation of the transitions imposed by the initial set. This is illustrated in figure 1. The set of initial pixels 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. The influence of the amplitude values of the upper edge (dashed line) at pixel x is 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 because the weights of the pixels located on the other side of the transition, at a larger geodesic distance, will be much smaller. As a result, the use of the geodesic distance allows the preservation of the transitions.

B. Two-step iterative algorithm

Starting from the set of initial pixels the *morphological interpolation* technique is implemented by an efficient two-step algorithm. The two steps, namely geodesic propagation step and smoothing step, are successively iterated until convergence.

- *Geodesic propagation step*
  In the *geodesic propagation step*, instead of computing geodesic distances from all the unknown pixels to every point of the initial set, the amplitude values of the known pixels are propagated by geodesic dilation to fill the empty areas of the image. This is performed by a fast algorithm, using FIFO queues, so that each pixel is treated only once in a complete propagation process. Figure 2 shows some intermediate images corresponding to the propagation process from a given initial set (a synthetic initial image consisting of two small geometric figures).

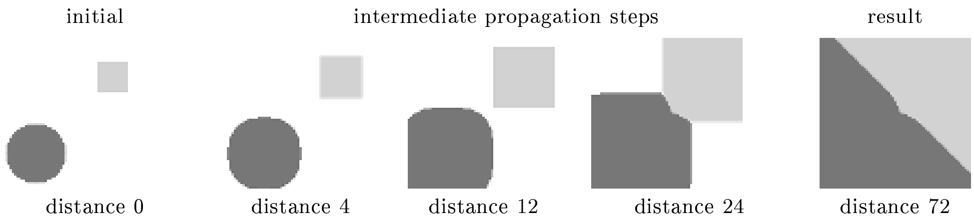


Fig. 2. Geodesic propagation: initial pixels and some intermediate images

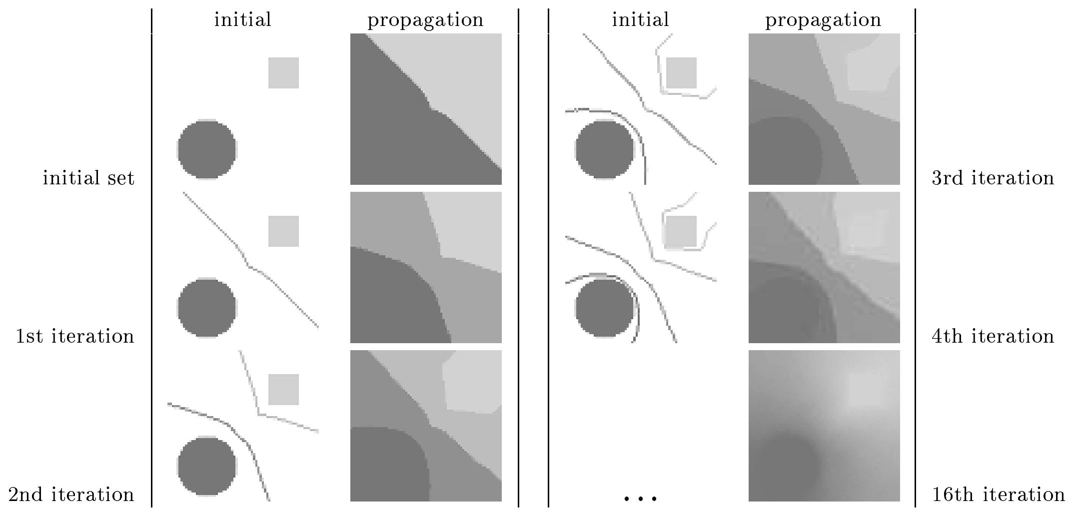


Fig. 3. Smoothing iterations: left images, initial and secondary pixels; right images, propagation

- *Progressive smoothing step*
  At the positions where two or more propagation fronts originated from initial pixels of different amplitudes meet, the process stops and a false transition is created. The false transitions appearing outside the set of initial pixels are smoothed in the second step. The morphological Laplacian[1] is used as a transition detector in order to obtain these false transitions. Pixels on both sides of the false transitions compose the set of *secondary pixels.* A grey level value equal to the average of the intensity values on both sides of the transition is assigned to each secondary pixel. This is the *smoothing step.* Secondary pixels will be used in the next iteration of the algorithm in order to smooth out these transitions.
- *Iteration*
  Then, a second iteration is performed: the propagation step propagates the grey level values from the sets of initial as well as secondary pixels. The propagation creates new false transitions which define a new set of secondary pixels where grey 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 1) propagation of values from the initial and secondary pixels, and 2) smoothing of the grey levels at the false transitions, is iterated until idempotence. Figure 3 illustrates several iterations of the algorithm. Please observe the progressive smoothing of the false transitions. After a few number of iterations, the algorithm quickly converges to the final interpolated image.

C. Algorithm efficiency

The efficiency of the morphological interpolation algorithm in terms of computational load is illustrated in table II-C. Comparative figures of execution time[2] are given for the previous example of morphological interpolation and solved by applying linear diffusion by means of iterated space variant filters. The result is very similar in both cases, as can be observed in figure 4. Please notice the drastic reduction in the number of iterations needed for the morphological 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, decreasing 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 of multigrid techniques for speeding up convergence when the morphological interpolation algorithm is used.

[1]The morphological Laplacian, $L(f)$, is defined as the residue of the gradient by dilation, $g^+()$, and the gradient by erosion, $g^-()$:

$$g^+(f) = \delta(f) - f \tag{1}$$

$$g^-(f) = f - \varepsilon(f) \tag{2}$$

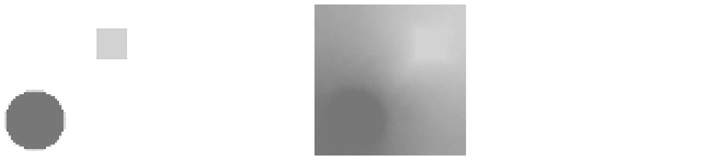
$$L(f) = g^+(f) - g^-(f) \tag{3}$$

The morphological Laplacian 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. Indeed, it can be shown that the morphological Laplacian is an approximation of the signal second derivative.

[2]Note: CPU times were computed on a Sun SPARC10 workstation

Interpolation technique	linear diffusion	multigrid diffusion	morphological
Execution time (sec)	312,8	53,3	2,8
No. of iterations	4980	equivalent to 795	16

TABLE I
Execution times of morphological interpolation vs interpolation based on diffusion equations for the previous example



initial set morphological interpolation linear diffusion

Fig. 4. Comparison of morphological interpolation and linear diffusion results

III. Application to "sketch-based" image coding

A smooth approximation of an image may be obtained by interpolation from the set of pixels with large curvature values. The following experiment has been carried out in order to investigate its possible application to image compression. In the left image of figure 5, a set of pixels having large absolute values of the second derivative (actually, the morphological Laplacian) is shown. If we attempt to interpolate the remaining pixels of the smooth areas in between, the result will be the one presented in the right image. About one tenth of the pixels of the image have been used as initial points for the interpolation algorithm. The peak to noise ratio of the interpolated image of figure 5 is only 23 dB but its subjective quality is not bad, because our attention is primarily drawn to the strong transitions which have been correctly placed and reproduced.

This experiment proves that it is possible to obtain a fair approximation of the original image from the amplitudes and positions of some pixels having large curvature values. The technique described in the previous section has been used for the interpolation process. Furthermore, the morphological Laplacian performs as an effective enhancement operator for the detection of such set of initial pixels. Obviously, the application of this idea to image coding relies on the selection of a proper set of initial pixels for the interpolation process. The *initial set* should lead to a compact representation and, at the same time, allow a good approximation of the original image by interpolation.

A. Image coding by maximum and minimum curvature lines

The extrema of the second derivative locate the points with largest curvature values. The lines of largest curvatures are placed at the upper and lower side of each transition, bringing information about the transition width and the intensity change. These lines are called upper and lower *edge brims* by some authors [10] and may be obtained as

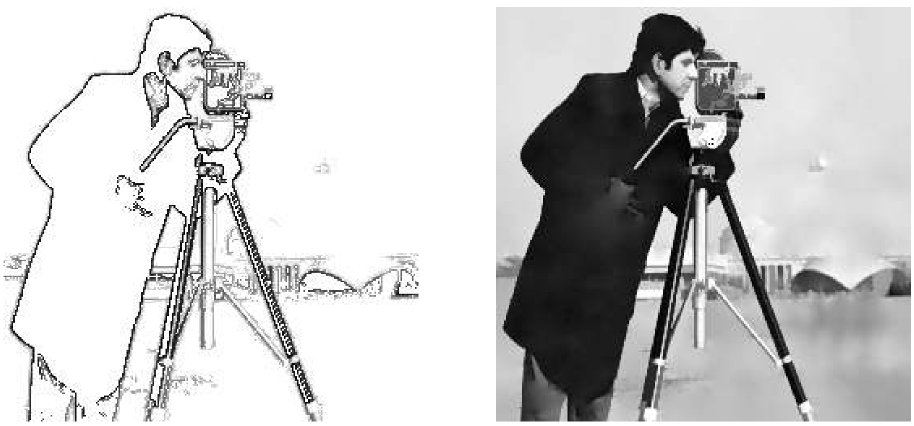


Fig. 5. Morphological interpolation from pixels with large Laplacian values: left, initial image (about 10% pixels); right, interpolation result

Fig. 6. Interpolation from lower and upper edge brims: left, brims' positions; right, interpolation result at 0.18 bpp

the 'crest' and 'valley' lines of a second derivative operator. Edge brims do look promising for the characterization of visual information from a perceptual point of view. Robinson [12] claims that brim lines are less noisy than Laplacian zero-crossings, which follow the edge midpoints, and have been often used for contour extraction. Edge brims do not show so many random fluctuations because they do not represent a very rapid change in value with respect to position as edge midpoints do.

In the left image of figure 6, the white and black lines correspond, respectively, to the crest and valley lines of the Laplacian or, likewise, to the positions of the lower and upper edge brims of the initial image. Edge brims may be detected by computing the watershed [9] of the Laplacian and of its dual with an appropriate set of markers. In order to obtain the lower brims (crest lines of the Laplacian), the set of markers is formed by the union of two sets: the flat areas of the original image larger than a given size and the connected components of negative Laplacian values indicating the presence of valleys. For the upper brims (valley lines of the Laplacian), the second set is formed by connected components of positive Laplacian values indicating peaks and ridges. Please notice that some pieces of contour have been removed from the watershed result, either because the Laplacian was not significant enough at these positions or because the lines were too short. The necessary thresholds have been chosen on an empirical basis.

The geometric structure of the brim lines may be coded at low cost by means of a contour-following technique. The amplitudes of the initial pixels in these lines should also may be coded with a few number of bits. Given that intensity values along the edge brims should keep rather constant, some approximation may be employed to code the values within each brim line. If the initial set is composed of the pixels at the positions indicated by the watershed lines shown in figure 6 with the approximated intensity values, the interpolation results in the right image of the same figure.

B. Two-component model

We put forward a two-component model for perceptual image coding that strongly relies on morphological operators. The interpolation result of figure 6 corresponds to the *primary* component of the perceptual model. The residual component, or *texture* component, contains the fine textures, which will be separately coded by a subband coding scheme.

- *Primary component*
  The first component of the model consists of the strong edges and smooth areas of the image. The smooth areas are generated by interpolation from the positions and amplitudes of the pixels of the initial set, i.e. the lower and upper brims of strong edges. A derivative chain code technique [7] is used to code the pixels' positions, whereas the amplitude values have been coded by polynomial approximation. More precisely, the network of brim lines is broken at each triple point (points with more than two branches). Then, the amplitudes of the pixels located under the resulting curves are approximated by a first order polynomial. The two coefficients defining each polynomial are quantized, entropy coded and transmitted. In the example of figure 6, the overall bit-rate needed for the primary component is 0.18 bits per pixel. About 16% of this rate is spent in the coding of amplitudes, 70% for the chain-code information and the remaining 9% for the initial positions of each brim line.
- *Texture component*
  The coding residue of the first component –computed as the difference between the original image and the interpolation result– mainly consists of fine textures. This second component of the model is shown in the left image of figure 7. It lacks of significant transitions and may be approximated by a waveform coding technique. A coded reconstruction at low bit-rate (0.15 bpp) is shown in the right image of the same figure. It has been obtained by the application of the linear subband coding scheme presented in [2]. Information about the edge structure –available from the first component– is used for the texture coding of the second component, so that the masking effect of

Fig. 7. Texture component: left, coding residue; right, subband coded texture at 0.15 bpp

strong transitions may be considered. Amplitude errors in the neighborhood of these areas are less noticeable for the human eye than in other parts of the image [3]. Therefore, the quantization process is allowed to introduce large errors near the transitions by employing adaptive quantizers and bit allocations over arbitrarily shaped sub-edge regions in order to reduce the total number of bits.

IV. Results and Conclusion

The compression ratio achieved with the above strategy is equal to 24 (0.33 bpp) for the addition of the strong edge and the fine-texture components of figures 6 and 7. The result is shown in the right image of figure 8. For comparison, the application of the JPEG standard [14] at the same bit-rate is shown in the left image of the same figure. The subjective quality of the described technique is significantly better because of the good rendition of the strong edges. The PSNR value (25.5 dB) is also larger than for the JPEG reconstruction (24 dB). The artifacts produced by the block-based DCT coding –blockiness in smooth areas and ringing in the neighborhood of strong transitions– are not present in the result of the two-component coding scheme. However, a different kind of visual artifacts may be observed. A certain smoothing effect is visible in some parts of the image and there are some missing objects, for instance the neck of the shirt has been almost removed.

The separate coding of strong edges permits the adaptation of the coding scheme to the visual perception of the images, avoiding unnatural degradations produced by waveform coding techniques at high compression ratios. A number of different artifacts are introduced by this method at low bit-rates. It is hoped that such effects are more naturally perceived than those of waveform coding techniques by the subjective judgement of the observer.

Mathematical Morphology provides powerful operators to perform shape analysis. Morphological operators are very useful for the detection of edge features in 'perceptually motivated' Second Generation image coding applications, as has been shown in the present paper. The morphological operators involved in the coding of the primary component, i.e. the watershed and the morphological Laplacian, perform very efficiently compared to more conventional techniques for edge extraction, like the LGO operator, or the diffusion filters iteratively applied for the interpolation of smooth areas reported in previous works. The new morphological interpolation technique intended for the scattered data interpolation, that has been described in section II has proven to be faster than linear diffusion techniques in order to generate the smooth component from the extracted edge features, with similar quality of the interpolation results.

References

[1] S. Carlsson. Sketch based coding of grey level images. *EURASIP, Signal Processing*, 15(1):57–83, July 1988.

[2] J. R. Casas and L. Torres. A feature-based subband coding scheme. In *ICASSP'96*, volume IV, pages 2357–2360, Atlanta, GA, May 1996.

[3] T. N. Cornsweet. *Visual Perception*. Academic Press, New York, 1970.

[4] P. Grattoni and A. Guiducci. Contour coding for image description. *Pattern Recognition Letters*, 11:95–105, February 1990.

[5] N. Jayant, J. Johnston, and R. Safranek. Signal compression based on models of human perception. *Proceedings of the IEEE*, 81(10):1383–1421, October 1993.

[6] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second generation image coding techniques. *Proceedings of the IEEE*, 73(4):549–575, April 1985.

[7] F. Marqués, J. Sauleda, and A. Gasull. Shape and location coding for contour images. In *Picture Coding Symposium*, pages 18.6.1–18.6.2, Lausanne, Switzerland, March 1993.

[8] D. Marr. *Vision*. Freeman, New York, 1982.

[9] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(4):21–46, September 1990.

[10] X. Ran and N. Farvardin. A perceptually motivated three-component image model. Part I: Description of the model. *IEEE Transactions on Image Processing*, 4(4):401–415, April 1995.

[11] X. Ran and N. Farvardin. A perceptually motivated three-component image model. Part II: Application to image compression. *IEEE Transactions on Image Processing*, 4(4):430–447, April 1995.

[12] J. A. Robinson. Image coding with ridge and valley primitives. *IEEE Transactions on Communications*, 43(6):2095–2102, June 1995.

[13] P. Soille. Spatial distributions from contour lines: an efficient methodology based on distance transformations. *Journal of Visual Communication and Image Representation*, 2(2):138–150, June 1991.

[14] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, April 1991.

Fig. 8. Results: left, JPEG at 0.35 bpp; right, described technique at 0.33 bpp