A Reprint from the

PROCEEDINGS

Volume 594

Image Coding

4–6 December 1985
Cannes, France

Adaptive split-and-merge for image analysis and coding

R. Leonardi, M. Kunt
Signal Processing Laboratory - Swiss Federal Institute of Technology, Lausanne
16, ch. de Bellerive - CH-1007 Lausanne, Switzerland

Adaptive Split-and-Merge for Image Analysis and Coding

R. LEONARDI and M. KUNT

Signal Processing Laboratory - Swiss Federal Institute of Technology, Lausanne
16, ch. de Bellerive - CH-1007 LAUSANNE (Switzerland).

Abstract

An approximation algorithm for two-dimensional (2-D) signals, e.g. images, is presented. This approximation is obtained by partitioning the original signal into adjacent regions with each region being approximated in the least square sense by a 2-D analytical function. The segmentation procedure is controlled iteratively to insure at each step the best possible quality between the original image and the segmented one. The segmentation is based on two successive steps: splitting the original picture into adjacent squares of different size, then merging them in an optimal way into the final region configuration. Some results are presented when the approximation is performed by polynomial functions.

Introduction

In the context of image analysis and coding, various studies[1-5] have shown the utility of decomposing the original 2-D signal $g(u,v)$ into a set of contiguous regions. This process is called segmentation. Among its various applications, source coding and redundancy reduction is one of the most important ones, and it is the concern of this paper.

Early work on the use of the segmentation for coding has led to rather high compression ratios[2]. This segmentation is based on a simple region growing procedure consisting in the approximation of 2-D image data by constant levels (zero-order approximation). Given the good results that were reached, a strong motivation was thus available to go further ahead. Unfortunately, the straightforward generalization of region growing with higher order approximations is very cumbersome. That is why a different approach is introduced to achieve the segmentation. It is based on a concept called adaptive split-and-merge. In the first step, the original image is divided iteratively into a set of squares of various sizes. Image data are approximated over each square. The procedure stops when a quality criterion is reached. In the second step, adjacent squares are merged if their joint approximation is satisfactory. In addition, this split-and-merge is a method quite similar to human behaviour in analysing a natural scene. It starts from a global look at the entire image, then goes to local areas to examin details. Finally, different parts of the image are associated together on the basis of some similarity measure.

In the following sections, parts that are discussed are the generalities, the approximation, the split process, the merge process, the post-processing and the final comments.

The problem

In this section, the problem is defined in general terms and requirements for the segmentation are stated.

The problem is to obtain a compact representation of the information content in a given image. Among several possible ways to do it, the general contour-texture description seems to be the most satisfactory[2,4]. The goal to reach is to describe the content of an image in terms of regions whose frontiers are the borders of the objects contained in the image. Of course, this ideal goal can only be approached.

According to the properties of the human visual system[4], if a certain degradation appears in describing the natural scene by means of an approximation, it is essential to mantain an accurate description of the frontiers of the different regions. In the context of image coding applications, the texture within each zone may be reproduced with some degradation as most of the semantic information is contained in the contour location.

The method of segmentation presented here should therefore meet the following requirements:

- the image is segmented into a set of different zones corresponding to slowly varying picture data,
- the number of such zones should be minimized, without destroying however real contours in the original scene,
- the error between the approximated signal and the original one within each zone should be kept below some threshold to insure sufficient visual quality,
- the resolution obtained to represent the borders of the segmented regions is the pixel size.

Approximation characteristics

In this part of the paper, different aspects regarding the approximation are analysed. Essentially, these include the error criterion used, the type and the number of approximating functions.

One section is devoted to this problem as any segmentation is highly dependent on the approximation process used to represent each region, and vice-versa. Therefore, given a set of r approximating functions $\Psi_i(u,v)$, there exists an optimal way of segmenting the original signal $g(u,v)$ with this set of approximating functions. As presented in a previous paper[3], the best least square (L^2) approximation $\hat{g}(u,v) = \Sigma\ \alpha_i \Psi_i(u,v)$ is computed with to the current image region. This region defines the domain $D_i = \{(k_i,l_i);\ i=1..N\}$ containing N pixel locations, on which the 2-D signal takes the set of N values $g(k_i,l_i)$. (N can be different from region to region). The coefficients α_i are choosen in order to minimize the least square error within the region. The best approximation in the least square sense has been choosen as :

- It is not sensitive to local distorsions in the set of data to be approximated. Even if such distorsions may correspond to real contours in the image, error measurements along such contours will allow to take another partitioning of the image (subdivision in case of split process, non association in case of merge operation). Therefore, the L^2 approximation will guarantee a good fit of most of the pixel values.
- A unique solution does exist to the problem as long as the functions $\Psi_i(u,v)$ are linearly independent and N is greater or equal to r. No restriction is made regarding the number of regions, their shape or the pixel values within each region.
- It is easier to compute than other types of approximations (such as minimax).

Let us now consider the problem of the type and number of approximating functions. The only restriction regarding the type of these functions is that they must be linearly independent. There are many advantages in using polynomials to represent slowly varying surfaces[3]. Qualitative motives related to the similarity of shapes between 2-D polynomials and portions of natural scenes suggest to choose the polynomial degree in increasing order.

As far as the number of approximating functions r is concerned, we shall keep it relatively small in order to map by transformation the approximated signal into a small number of coefficients α_i. In addition, when using polynomials, if their degree is too high, it is difficult to prevent the calculation of the L^2 approximation from round-off errors.

Orthogonal functions have not been considered so far, as it is difficult to define them on 2-D domains of varying shapes. Besides, there are good reasons to believe that the approximation will present oscillatory effects (see Fourier, Hadamard) visually very unpleasant if the least square error is not zero.

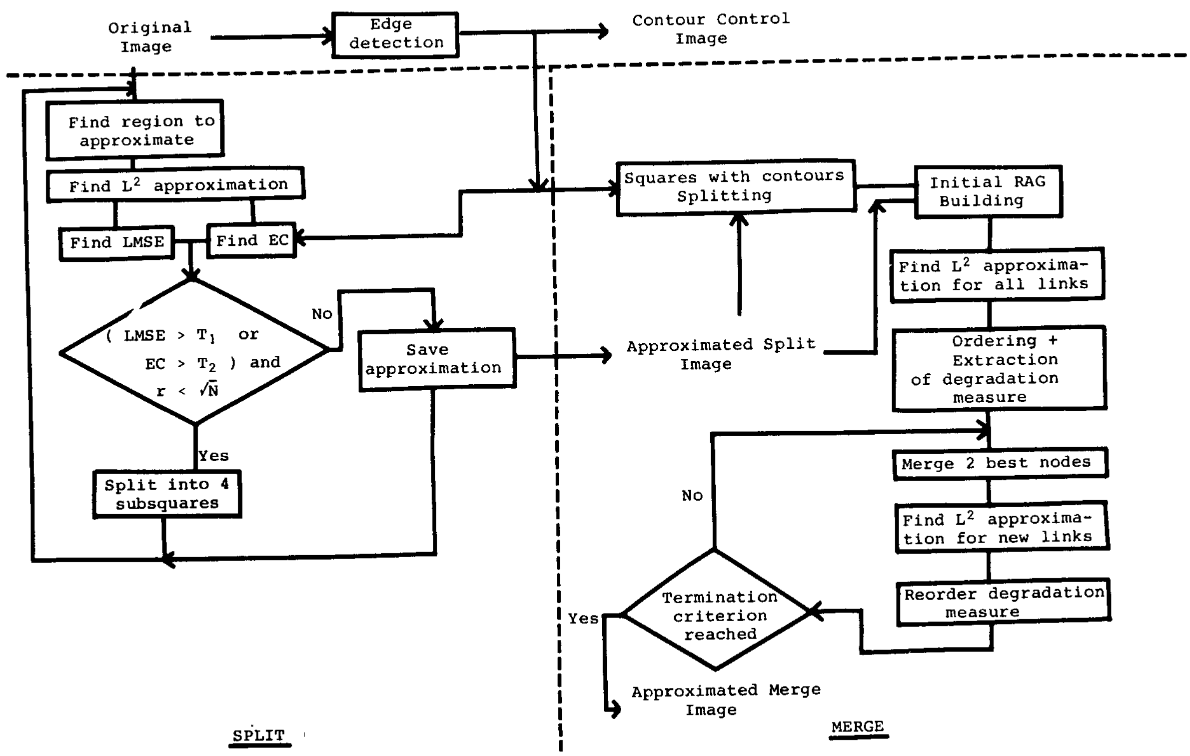


Figure 1 : Segmentation process.

Figure 1 presents the overall segmentation process that will be discussed now.

Split process

This section describes briefly the split operation with particular attention given to the error measurements used to control the segmentation. Results are presented when applying this split process with different types of polynomials.

An attractive solution for describing the image corresponds to represent the signal within each region with an approximation. The accuracy of the representation is defined with respect to the application for which the segmentation is used. For instance, no loss of information may be allowed in the context of medical X-ray photography storage. On the other hand, just one characteristic feature may be sufficient to classify the analysed region with respect to some recognition problem.

To control the overall segmentation process, a control image of the contours is extracted from the original image using any edge operator. Preferably, we have choosen the directional edge operator[6] or Marr-Hildreth's one[7] as they do take into account of the human visual system properties.

For each region on which the best approximation in the least square sense will be evaluated, we extract two indices of quality. The first one is a global measure represented by the least mean square error on the region :

$$\mathrm{LMSE} = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{g}(k_i,l_i) - g(k_i,l_i) \right|^2 \qquad (1)$$

where $\hat{g}(k_i,l_i)$ is the approximated value for the pixel location (k_i,l_i).
This measure is used to evaluate the quality of the approximation on a slowly varying area.

The second index of quality is based on the measure of errors at contour locations within the region of interest :

$$\mathrm{EC} = \frac{1}{N} \sum_{i} \left| \hat{g}(k_i,l_i) - g(k_i,l_i) \right|^2 \qquad (2)$$

(k_i,l_i) D, with (k_i,l_i) being a contour point.

Whenever EC or LMSE exceed a threshold, the region of interest is not sufficiently well approximated so that the initial partitioning of the image must be modified.

For understanding ease, the number of sampling points of the digital picture is a power of 2, let say 2^{2q}. Starting then with an initial square $2^q x 2^q$, its L^2 approximation is evaluated with the set of functions $\Psi_i(u,v)$. Whenever LMSE or EC are beyond their respective acceptance threshold, the initial square is split into four squares of identical size $2^{q-1}x2^{q-1}$. The same procedure is iterated for every subsquare until the quality measure becomes satisfactory. This split process is insured to stop since the set of Ψ_i functions will certainly interpolate the square pixels as soon as r exceeds N. In practice and to avoid infinite solutions for the interpolation, the process is stopped as soon as r is greater than $2^{2*(q-l-1)}$, where l is the last level of split. As long as r is sufficiently small, practical experience has shown that EC and LMSE are kept low enough, just at the last level of split. We shall see how this will not deteriorate the optimal reconstruction in the merge process.

As presented in the introduction, this part of the segmentation allows to pass from a global analysis of the image to a local one as long as details are present. A great correspondance between the original signal and the approximated one can be obtained whenever a precise control image of the contours is used.

Computational aspects of this adaptive split method were presented in a previous publication[3]. It was also described how the problem could be simplified whenever a separable formulation of the problem could be used.

After the segmentation procedure, the 2-D signal is represented by :

1) the location of the different segmented regions,
2) the approximation within each region.

In this split process, the shapes of the segmented regions are squares of different size. By taking into account geometrical constraints, the structure of the split graph can be reduced to a quadtree representation[8].

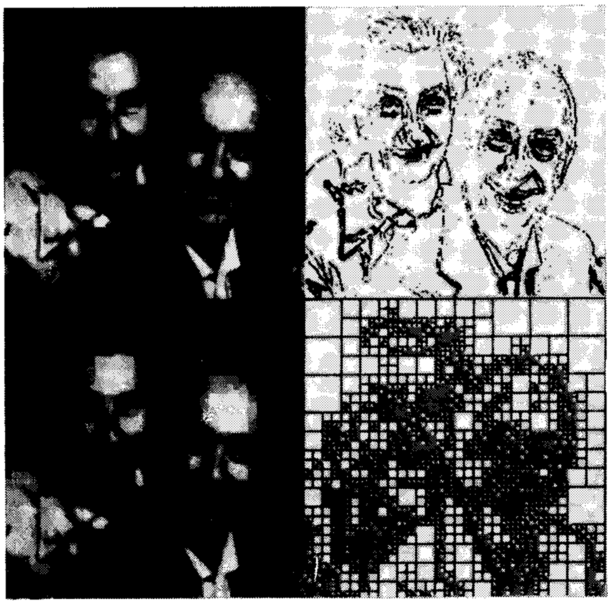
It is difficult to estimate the necessary quantization to represent every coefficient α_i, because it is very much dependent on the type of functions $\Psi_i(u,v)$ used. In order to have an estimation of the coding efficiency of the split algorithm, a quick code was derived for the zero-order approximation ($\Psi_1=1$). The coded coefficient is insured to be represented with atmost eight bits if the original signal has 256 quantization levels. Compression ratios of the order of 30:1 to 60:1 have been obtained with this technique for the overall adaptive split process (see figure 2e).

When using more approximating functions, it is important to anlyse the relevance of using them all in representing the approximation. (A constant luminance zone does not need to be represented by three coefficients corresponding to an approximation by planes). As As an indication[5], 12 bits per coefficient were sufficient in the case of a quadratic approximation.

Examples of the split process are presented for two different pictures on figures 2 to 4. For figure 2, we have applied the split procedure with a zero-order approximation to the portrait picture, resulting in one coefficient per region and a total number of 5335 regions (the smallest size square is limited to 2x2). In figure 3, the portrait is approximated by a set of planes within each region. The corresponding segmentation involves then three coefficients per region and 4615 regions. Figure 4 gives the result of the split process for the building picture when using a first-order separable polynomial (first degree in u and v along both directions). Four coefficients are necessary to represent each region and the total number of such regions is 6655.

a) b)

c) d)



e)

Fig. 2 : Portrait picture with a zero order approximation (5335 reg.)

2a) original image
2b) contour control image
2c) split approximated image
2d) mosaic of squares after split
2e) coding + enhancement of image 2c) (compression ratio : 49.7 to 1)

a) b)

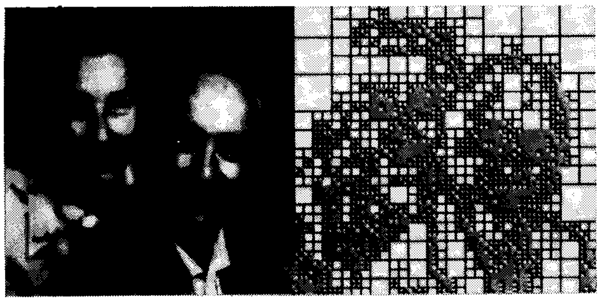
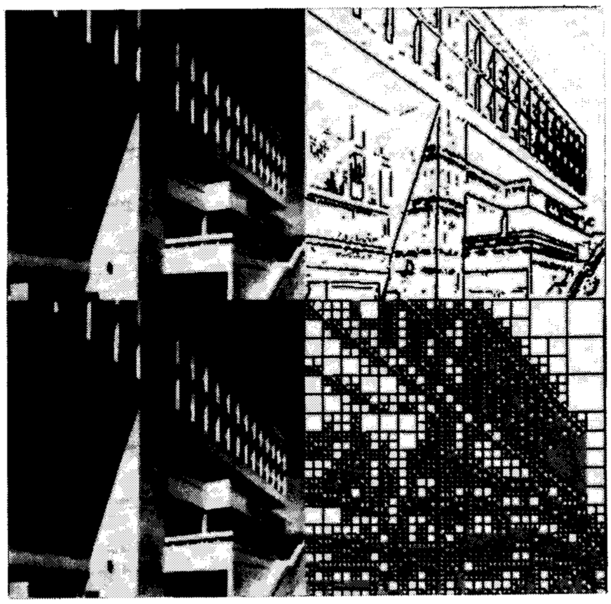


Fig. 3 : Portrait picture with an approximation by planes (4615 regions)

3a) split approximated image
3b) mosaic of squares after split

a) b)

c) d)



4a) original image
4b) contour control image
4c) split approximated image
4d) mosaic of squares after split

Fig. 4 : Building picture with a first order separable approximation (6655 regions - $\Psi_1=1$, $\Psi_2=u$ $\Psi_3=v$, $\Psi_4=uv$)

Merge process

This section will explain how it is possible to associate different regions obtained by the split operation in order to get a better segmentation of the original image. Applications of this procedure to the portrait and building pictures are presented. The effect of various quality measurements used to control the segmentation in an optimal way is also discussed.

The merge process was derived from an adaptive region growing technique[5]. Any segmentation algorithm requires the definition of an appropriate data structure in order to effectively access and relate the different regions. The data structure chöosen to merge various squares obtained by the split algorithm, is the region adjacency graph (RAG)[9]. This is a classical map graph with each node corresponding to a region and links joining the nodes representing adjacent regions. Only contiguous regions are considered as these should be associated first to insure the connexity of the final segmented regions.

The basic idea of the merge algorithm corresponds, first, to assign to every link in the graph a value representing the "degree of dissimilarity" that exists between nodes that this link connects. This degree of dissimilarity constitutes a quality measurement of the approximation. In a second step, the link that exhibits the lowest degree of dissimilarity is removed and the nodes it connects are merged into one. The procedure is iterated until a termination criterion is verified. At each merging step, the values associated to the links that previously connected the two nodes to the rest of the graph are recomputed.

Two termination criterions were considered to stop this part of the segmentation :

- the minimum number of regions of the segmented image. This way, one can control the data compression aspect, by knowing the average cost to represent each region.
- the maximum acceptable dissimilarity between the original image and the approximated one. This can be analysed locally if we desire a minimum quality for each region or globally as, for instance, the total square error on the entire image. This termination criterion allows to limit the degradation of the approximated image.

Thus, we have defined a method which guarantees the best possible segmentation with respect to the choosen approximating functions $\Psi_i(u,v)$, the initial RAG and the quality measurement to control the segmentation. Regions are sequentially merged together only where they cause the slightest increase of error in the image approximation.

The initial RAG is essential as it will allow to reach a precision of the order of the pixel size in the region frontier location. In fact, this is impossible if we use exactly the same graph obtained after the split process except in the case of a zero-order approximation. To reach an optimal resolution in describing the region borders, the split step should divide the original image all the way to the pixel size. As there exists an infinite number of interpolations with the set of $\Psi_i(u,v)$ functions whenever r is greater than N, it is meaningless to associate regions with less than r points. To put into correspondance region frontiers with real contours, it is therefore adequate to associate regions that do not contain contours to the smallest possible parts of regions who do have contours. A good way of doing this is to divide after the split operation, regions with real contours into all their pixels, which define then initial regions in the initial RAG. A constraint must be inserted to the merge process itself: regions that define one pixel only, cannot be merged together but only to regions having more than r points. The merge process may then stop with some contiguous pixels which could not be inserted in bigger regions but which represent slowly varying areas when associated to each other. In such a situation, an adequate pixel association process must be derived to create a unique region for all these pixels. The number of remaining regions can be, this way, highly reduced.

To control the segmentation process, various measures of dissimilarity were derived. The LMSE within each region was considered. The total square error within each region, defined as TSE = LMSE*N was also taken into account. The first measure tends to associate small regions into much bigger ones even if they belong to different objects of the original scene. In fact, small regions induce only a small change in the approximation when associated to big ones. The error within themselves does not significantly increase the total LMSE as their error is divided by the total number of points of the two regions.

On the contrary, the TSE tends to associate small regions together even if they belong to different textures. This may be useful in coding applications if we allow to lose small elements. It should be changed as soon as small remaining regions can no more be considered superficial details.

On the other hand, if we want to keep all the details, it is possible to define a measure of dissimilarity which is independent of the region size. This is obtained by weighting within each region the measure of error by their corresponding number of points. This error is denoted by ER and it is defined by :

$$ER = \frac{1}{N_1} \sum_{i=1}^{N_1} \left| \hat{g}(k_i,l_i) - g(k_i,l_i) \right|^2 + \frac{1}{N_2} \sum_{i=N_1+1}^{N_1+N_2} \left| \hat{g}(k_i,l_i) - g(k_i,l_i) \right|^2 \quad (3)$$

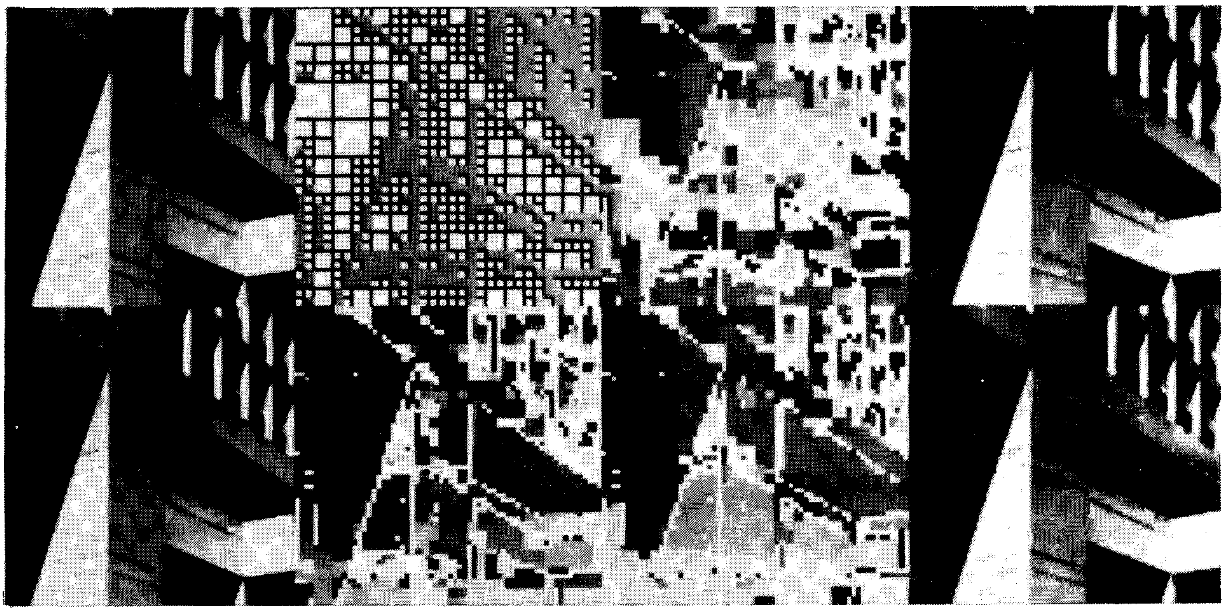
where N_1 and N_2 corresponds respectively to the number of pixels of each node connected by the considered link in the RAG.

Examples of the effect of these different measures of dissimilarity are shown in the sub-images of figure when planes are used to approximate the 2-D signal. Figures 6 and 7 present the split-and-merge result when applied to the building picture and the portrait one respectively. The approximation is again performed with planes ($\Psi_1=1$, $\Psi_2=u$, $\Psi_3=v$). The dissimilarity measure used to control the merge process is the LMSE. From the split segmentation, we have reduced the number of regions from 6688/4615 to 999/999 respectively, leading therefore to a great redundancy reduction. The degradation is not significant before and after the merge processing. It is also interesting to notice the improvement of quality when comparing these results to those obtained with a zero-order approximation and the region growing technique[2] (pp. 103,106), the number of regions being approximatively the same.

At this stage of our work, computation time has limited the merge operation to all squares that did not contain contour points. The other regions are not included in our account. It is reasonable to assume that in most of the cases, the contour points are inserted into much bigger neighbouring regions, so that the final number of regions will remain unchanged.

b) c)

a) d)



e) h)

f) g)

Fig. 5 : Sub-image of the building picture approximated by planes.
5a) split approximated image (1840 squares) 5d) TSE merge approximated image (300 reg.)
5e) LMSE merge approximted image (300 reg.) 5h) ER merge approximated image (300 reg.)
5b), 5c), 5f), 5g) : mosaic images of the regions corresponding to 5a), 5d), 5e) and 5h).

a) b)

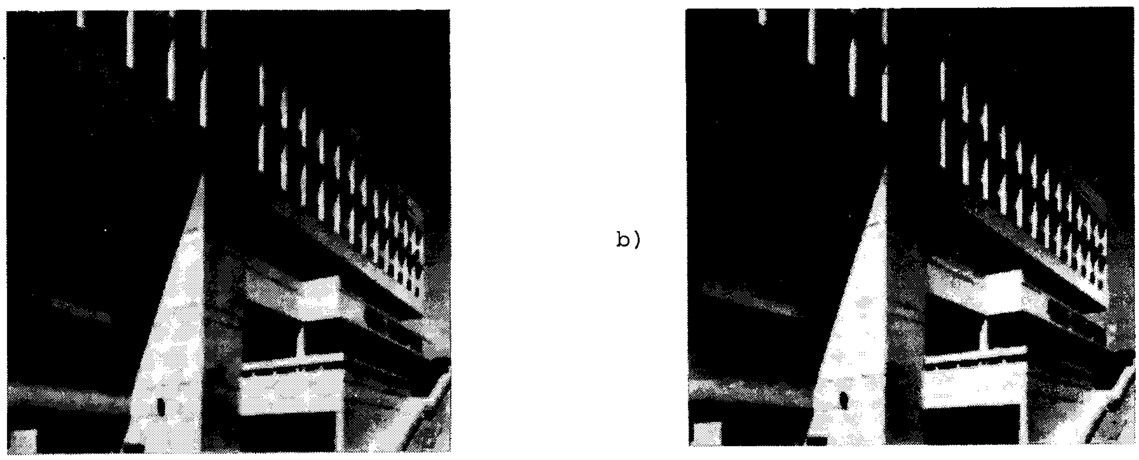


Fig. 6 : Building picture with an approximation by planes

6a) split approximated image (6688 squares).

6b) LMSE merge approximated image (999 regions).

a) b)

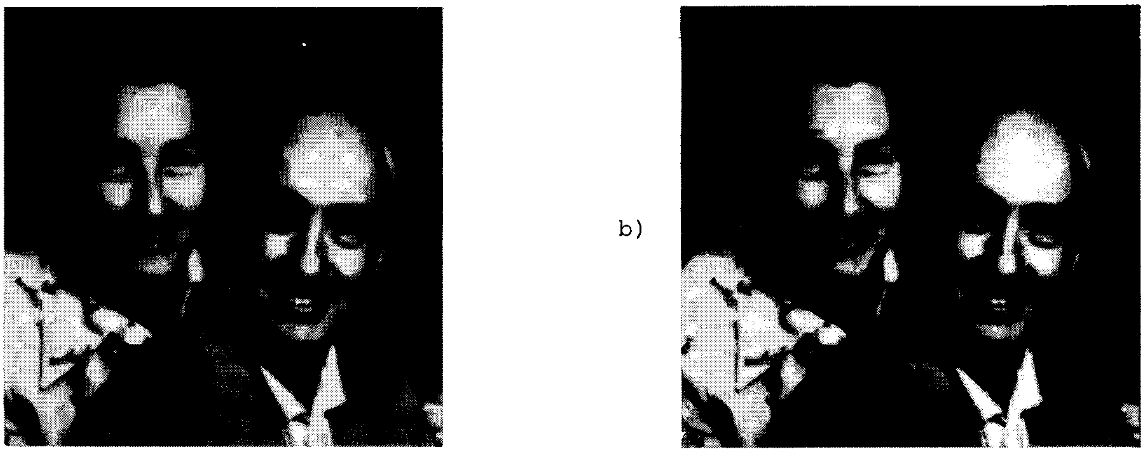
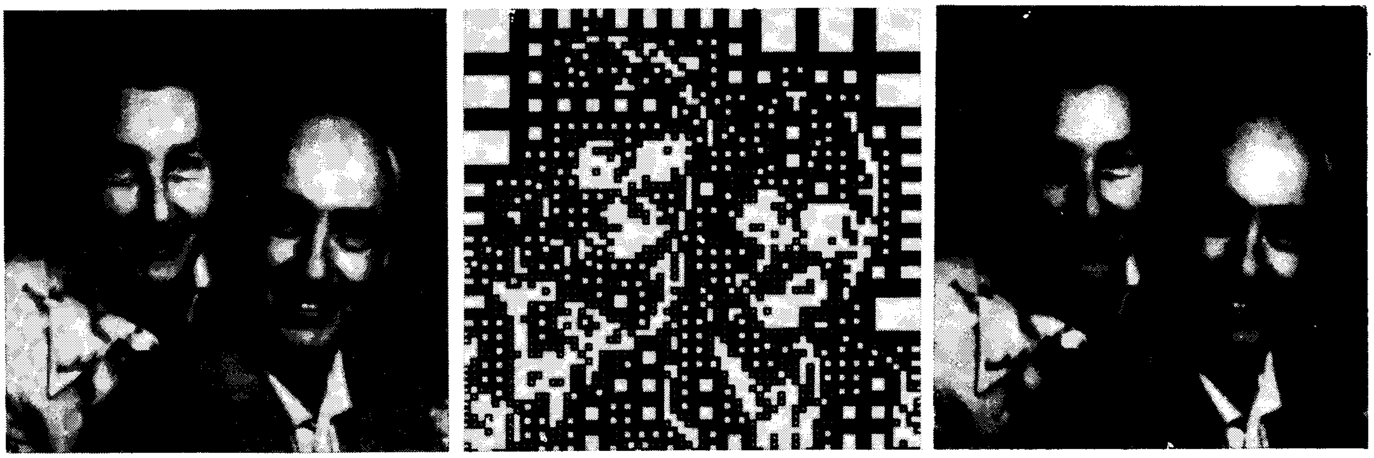


Fig. 7 : Portrait picture with an approximation by planes
7a) split approximated image (4615 squares).
7b) LMSE merge approximated image (999 regions).

Post-processing

We shall present here a smoothing technique that can be used to enhance the quality of the segmented image in case of a polynomial approximation. This procedure can be used after the split-and-merge algorithm. Due to the structure of polynomial functions, the approximated signal may be discontinuous between adjacent regions. This may create "false contours" at the boundaries of some adjacent regions. Between two consecutive crossing points of the region frontiers, just one bit is necessary to represent whether this portion of border corresponds to a false contour or not. For those who are interested in applying this technjust after the split process, let us notice that false contours can appear only on squares that have side greater than 2^{q-1} where l is the last level of split. The cost to represent then the regions to be smoothed can then be highly reduced. This can be generalized to those regions that have no inner points after the merge process. Once it has been established which contours do not correspond to real edges in the original picture, a smoothing algorithm is applied to both sides of this false contour. The width for which the approximated signal is smoothed with respect to each region, is linearly dependent on the number of points of the considered region. This smoothing filter corresponds to a low-pass such as moving average. Another solution was presented in a previous paper[5]. As an example, figure 8 shows the result of this post-processing to the image of figure 3b.



a) b) c)

Fig. 8 : Enhancement of the portrait picture approximated by planes
8a) split approximated image (=3b)
8b) image showing in black the regions that are smoothed
8c) enhanced image after smoothing

Conclusions

In this paper, an optimal segmentation algorithm was proposed for 2-D signals with respect to the set of approximating functions and the degradation measurements used to control the segmentation. The split process, when analysed by its own, gives already encouraging results with respect to coding applications. By reducing the number of segmented regions by afactor of 4.6 to 6.6 from the split graph to the final merged one, it seems reasonable to consider that the coding efficiency will be even more increased. Various parameters based on human visual system and understanding aspects are still under investigation to increase the quality of the segmented picture and to reduce the number of remaining regions without significantly affecting the semantic information present in the picpicture. These include essentially the definition of a better measure of dissimilarity to control the merge process and the implementation of the smoothing algorithm to enhance the quality of the merge result whenever false contours appear. A better measure for controlling the segmentation process should magnify the importance of errors at contour locations, whenever these edges define structured shapes. A more sophisticate error measure than EC may be necessary. Finally, a coding strategy should be defined to represent the approximated picture. If methods to describe region contours already exist[10], the problem of the quantization of the coefficients α_i must be anlysed. However, it is important to remind that most of the cost will appear in coding the position of the regions, rather than their texture, especially if it has been extremely simplified (for example, plane description).

Acknowledgements

Special thanks to M. Vetterli, J.-C. Gregoire, M. de Saint Denis and M. Kocher for the fruitful discussions we had and their help in finding solutions to computational aspects of the problem.

References

1. T. Pavlidis, "Segmentation of Pictures and Maps Through Functional Approximation", Computer, Graphics, and Image Processing, Vol. 1, No. 4, pp. 360-372.

2. M. Kocher, Codage d'Images à Haute Compression Basé sur un Modèle Contour Texture, Ph.D. thesis No. 476, Swiss Federal Institute of Technology, Lausanne, Switzerland, 1983.

3. R. Leonardi and M. Kunt, "Adaptive Split for Image Coding", Proceedings of the IASTED International Symposium on Applied Signal Processing and Digital Filtering, Paris, June 19-21, 1985.

4. M. Kunt, A. Ikonomopoulos and M. Kocher, "Second Generation Image Coding Techniques", Proceedings of the IEEE : Special Issue on Visual Communication Systems, Vol. 73, No. 4, April 1985, pp. 549-574.

5. M. Kocher and R. Leonardi, "Adaptive Region Growing Technique Using Polynomial Functions for Image Approximation", Signal Processing (to be published).

6. R. Cusani, "New Results in Directional Decomposation Based Image Coding", Proceedings of the SPIE Conference on Image Coding, Cannes, Dec. 1985.

7. D. Marr and E. Hildreth, "Theory of Edge Detection", Proccedings R. Soc., Vol. B207, 1980, pp.187-217.

8. Y. Cohen, M.S. Landy and M. Pavel, "Hierarchical Coding of Binary Images", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 3, May 1985, pp. 284-298.

9. T. Pavlidis, Structural Pattern Recognition, Springer Verlag, Berlin, 1977.

10. M. Eden and M. Kocher, "On the Performance of a Contour Coding Algorithm in the Context of Image Coding - Part II : Coding a Contour Graph", Signal Processing (To be published).