N90-16214
554738 p12

HIGH COMPRESSION IMAGE AND IMAGE SEQUENCE CODING

Murat KUNT
Signal Processing Laboratory
Swiss Federal Institute of Technology, Lausanne
CH - 1015 Lausanne, Switzerland

ABSTRACT

The digital representation of an image requires a very large number of bits. This number is even larger for an image sequence. The goal of image coding is to reduce this number, as much as possible, and reconstruct a faithful duplicate of the original picture or image sequence. Early efforts in image coding, solely guided by information theory, led to a plethora of methods. The compression ratio reached a plateau around 10:1 a couple of years ago. Recent progress in the study of the brain mechanism of vision and scene analysis has opened new vistas in picture coding. Directional sensitivity of the neurones in the visual pathway combined with the separate processing of contours and textures has led to a new class of coding methods capable of achieving compression ratios as high as 100:1 for images and around 300:1 for image sequences. This paper presents recent progress on some of the main avenues of object-based methods. These second generation techniques make use of contour-texture modeling, new results in neurophysiology and psychophysics and scene analysis.

INTRODUCTION

Every image acquisition system, be it high resolution microdensitometer or TV camera, produces pictorial data by sampling in space and in time, and quantizing in brightness, analog scenes. A digital image is thus an N by N array of integer numbers or picture elements (pixels) requiring N^2 B bits for its representation where B is the number of bits per pixel. This array is commonly referred to as the canonical form of digitized pictures. Generally, the canonical form requires a very large number of bits for its representation. For example, with a 512 by 512 raster and 8 bits per pixel, 2.10^6 bits are needed, a rather large number! For a 3 minute image sequence, at a rate of 25 images per second, the data rate becomes 50.10^6 bits/sec. The goal of image coding is to reduce (to compress), as much as possible, the number of bits necessary to represent and reconstruct a faithful duplicate of the original picture or image sequence. How high a compression can be achieved when a saturation has been reached within the framework of information theory and coding theory ? By simply going out of this framework with the so-called second generation methods [1]. A view of the difference between the techniques of the first and the second generation is the following. Image coding is basically carried out in two steps: first, image data is converted into a sequence of messages and, second, code words are assigned to the messages. Methods of the first generation put the emphasis on the second step, whereas methods of the second generation put it on the first step and use available results for the second step. The very end of almost every image processing system is the human eye. Although our visual system is by far the best image processing system one can think of, it is also far from being perfect. So, if the coding scheme is matched to the human visual system and attempts to imitate its functions, at least for the known part of it, high compressions can be expected. An image can be described in terms of several possible entities such as pixels of the canonical form, a group of pixels in small blocks, Fourier or other transform coefficients, linearly predicted values or derivatives, energy measures within a certain frequency band, etc. With the continuous progress in visual pattern recognition and scene analysis, another possibility is to describe an image in terms of contour and texture [2]. Two main avenues are followed in this paper. The first one imitates some neuronal processing using directional decomposition. The second is based on the segmentation of an image into regions so that region borders fit as much as possible contours of the objects, using either region growing or split and merge.

DIRECTIONAL DECOMPOSITION BASED CODING [3] - [5].

Directional filtering is based on the relationship between the presence of an edge in an image and its contribution to the image spectrum. It is largely motivated by the existence of direction sensitive neurones in the human visual system. A filter whose frequency response covers a sector or a part of a sector in the frequency domain is called a directional filter. To make edge detection with these filters easier, high-pass filtering along the principal direction is introduced. Areas of the Fourier domain corresponding to these filters are shown in Fig. 1. The entire frequency plane is thus covered with n directional filters and one low pass filter. The ideal frequency response of the i-th directional filter is given by

$$H(f,g) = \begin{cases} 1 \textit{ if } \vartheta(i) \leq \tan(\frac{g}{f}) < \vartheta(i+1) \textit{ and } f^2+g^2 \leq \rho_c^2 \\ 0 \text{ otherwise} \end{cases}$$

with $\vartheta(i) = \frac{i-1}{2\pi}$, $\vartheta(i+1) = \frac{i+1}{2\pi}$ and $|f|, |g| < 0,5$

where f and g are spatial frequencies, ρ_c is the cutoff frequency of the low-pass filter and where a unity sampling step size is assumed. Accordingly, a directional filter is a high-pass filter along its principal direction and a low-pass filter along the orthogonal direction. Because of the Gibbs phenomenon, the ideal frequency response of the directional filters should be modified by an appropriate window function. The purpose is to avoid oscillation around zero crossings corresponding to real edges. One of the most appropriate window functions for this purpose is the Gaussian window. After windowing the filters and filtering, the superposition of all the directional images and the low pass image, lead to the original image. Thus, the directional filtering, as defined, is an information preserving transformation. There are two parameters involved in the directional filters: their number and the cutoff frequency of the low-pass filter. The number of filters is directly related to the minimum width of edge elements that is accepted a priori in the image. Therefore, a direct way to define the number of filters (directions) is obtained by fixing the minimum length of accepted edge elements. From a physiological point of view [6], it seems that the quantization of the directions is made by 20 to 30 different groups of cells, each one specialized to a limited interval of directions. The choice of the cutoff frequency influences only the compression ratio and the quality of the decoded image.

The messages to be coded are the directional images and the low-pass image. Note that the following scheme is not information lossless and that a certain quality degradation is assumed when coding. The main objective is to achieve the highest compression for a given degradation. High frequency images will be used for detecting and coding edges. The loss of information comes from the inevitable choice between weak and strong edges. If the compression ratio is set to high rates, very weak edges must be eliminated. On the other hand, the indirect approximation of the edges by line segments, as assumed by the definition of the edge elements, introduces some degradations at the locations of high curvature. Edge detection in the directional images is based on the high pass character of the directional filters along their principal direction. Filtering a signal with a high pass filter gives zero crossings at the locations of abrupt changes (edges). Accordingly, edge detection in the directional images is performed by searching the zero crossings along the principal direction of each image. The strength of the edges to be retained is controlled by setting a threshold on their slope. Each directional image is represented by the positions and the magnitudes of the zero crossings. The positions are coded with run length coding using the Huffman code, requiring an average of 4.5 bits per position. The magnitudes of zero crossings are coded with three-bit code word.

The low frequency image can be coded in two equivalent ways. Since the maximum frequency of this component is much lower, it can be resampled using the two-dimensional sampling theorem and the resulting pixels can be coded by a standard procedure. The alternative is transform coding. The choice of the transform technique is directly dictated by the filtering that was used. The locations of the Fourier coefficients are known from the characteristics of the filter, and the importance of all these coefficients exclude any elimination by thresholding. This falls, therefore, in the category of zonal coding. After experimenting with several possibilities such as logarithmic quantization, bit allocation plane etc, the coefficients are quantized linearly. Fixed length words are used to code the phase and variable length words, as attributed by the Huffman code, are used to code the magnitudes.

In order to reconstruct the original image, all the components have to be decoded and added. The low frequency component is obtained by inverse transforming the coded coefficients. The high frequency component is obtained by synthesizingthe directional images from the zero crossings. The synthesis of edge profiles from the zero crossing information and the interpolation between the columns of the normalized directional images, are the most critical procedures for the quality of the decoded image. An edge model [1] offers the theoretical basis for the synthesis of the one dimensional signals along the edge directions. This model requires two parameters : the magnitude A of zero crossings representing the contrast of the edge and the standard deviation σ related to the steepness of the edge's slope. As the magnitude of zero crossings is coded, the only unknown parameter is the standard deviation. Experimental results indicates that a linear variation of the standard deviation with the contrast gives more realistic edges. The prototype wavelet which was adopted for approximating the profiles of zero crossings is the following:

$$g(u) = \frac{u}{Ak} \exp(-\frac{u^2}{Ak})$$

where u is the distance from the zero crossing at $u = 0$, A the magnitude and k a constant. Once the synthesis of zero crossing profiles is carried out at coded locations, the whole directional image is reconstructed by interpolation between the columns of the subsampled images. For a perfect interpolation between the columns of these images, the fact that the edge elements assumed by the presence of each zero crossing may have any direction within this interval must be taken into account. The interpolation algorithm consists in looking for a neighboring point not only on the same line but also on the two previous or next lines. A first series of decoded images with low compression ratio are shown in Fig. 2. The average compression ratio is around 50 to 1 and the quality of the picture is quite high. By decreasing the cutoff frequency and increasing the zero crossing detection threshold, a second series of results are obtained with higher compression ratios as shown in Fig. 3.

A new step can be made to decrease the redundancy of information by relating the directional images or edges to a prediction model [7]. The goal of this approach is to code the prediction coefficients and errors with less bits than the original information, which implies a good choice of the prediction structure. The study of the computational problems related to the solution of large linear systems for prediction error minimization leads to the conclusion that a synthetic model of the information in the directional image must be built to perform the prediction to avoid tremendous computation. Since in directional images the main information is concentrated in edges, a 2-D linear prediction is chosen. A vector is associated to each edge element, including its position and local profile parameters (magnitude and width) as its components. Then, the prediction model operates on these vectors and estimates edge position and parameters within a prediction structure defined on a set of connected edges.

SEGMENTATION BASED CODING[8]-[11].

In the first stage of this method, the image is segmented to classify its pixels into contour pixels and texture pixels. This procedure partitions the image into a set of adjacent regions under the constraint that the variation of the grey level within the region does not contain any sharp discontinuities, i.e. contours. Segmentation is carried out in three steps: preprocessing, region growing and elimination of artifacts. The preprocessing is intended to reduce the local granularity of the original image without affecting its contours, so that not too many small regions are obtained after region growing. The mechanism of region growing is the following. Regions to be extracted must be characterized with some property in the first step. The property might be, for example, the grey level of a pixel, the variation of the grey level, or the energy within a given frequency band. The selection of this property plays a very important role in the complexity of the method and in the exactness of the contours obtained after segmentation. Then, starting with a given pixel in the picture, its neighbouring pixels are examined to see whether they share the same property. If this is the case, that pixel is included in the region, and in turn, its neighbouring pixels are examined, and so on. When there are no more pixels left, connected to the region and sharing the same property, the procedure stops and restarts at any other pixel which is not included in the first region. The segmentation is complete when all the pixels of the picture are assigned to some region. The property used in our first attempt was very simple: it was a fixed grey level interval. Although it has a constant width, this interval is made adaptive by moving it up and down on the grey level scale in order to intercept the maximum number of pixels. This displacement is constrained, however, so that previously intercepted pixels always remain in the region. Unfortunately, because of the simple property used, the number of these contours is much higher that that of the objects in the original image. Two possibilties are available to overcome this problem : introduction of some distortions by eliminating insignificant regions and their contours, or the use of a more refined property. The first alternative relies on two heuristics: elimination of the small regions and merging weakly contrasted adjacent regions. Statistical analysis indicates that roughly 70 per cent of the regions have less than 15 pixels. To avoid the creation of holes in the image, these regions are included in one of their adjacent regions. To minimize the corresponding distortion, the enclosing region is chosen as the adjacent region whose mean grey level is the closest to that of the small region to be included. By observing areas of constant luminance gradient in the pictures, it can be noticed that they are subdivided into regions even though there is no real contour. This is due to the property used in region growing which divides the image into regions of fixed grey level dynamic range. The second possibility to decrease the number of regions is thus to merge together adjacent regions whose contrast is below a certain level. The contrast between adjacent regions is defined as the mean grey level difference calculated along their common border.

Contours obtained after segmentation are a part of the messages to be coded. A precise description of contours is essential for the human visual system. In this technique contour coding is carried out as follows. Since regions are closed, contour points along the border of two adjacent regions are described twice, once for each region. Prior to coding, these points are removed from one the regions to be described and coded only once. A new and refined code [12] is used requiring about 1.3 bits per contour point.

The missing part of the messages after contour coding is texture coding. It is carried out in two steps. In the first step, the general shape of the grey level in each region is approximated by a two-dimensional polynomial function. The order of the polynomial is determined as a function of the approximation error and of the cost involved in coding polynomial coefficients. A three dimensional view of these approximations is shown in Fig. 3. In this particular case, the best ('cheapest') approximation is obtained with a first order polynomial function. In the second step, the granularity removed with preprocessing is added back in the form of a pseudo-random noise to render the image more natural and less 'painted by numbers'. Fig. 4 shows the final state of the decoded pictures with compressions ranging from 26:1 to 44:1.

Unfortunately, the straightforward generalization of the above described region growing with a more complex property (higher order approximations) is very cumbersome. For this reason a different approach is introduced to achieve the segmentation. It is based on adaptive split-and-merge [13]-[15]. 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.

For each region on which the best approximation in the least square sense will be evaluated, two indices of quality are extracted. The first one is a global measure represented by the least mean square error over the region, whereas the second is based on the measure of errors at contour locations within the region of interest. These locations are extracted from the original image using a valid edge operator, whose result is a control image used to control the overall segmentation process. Assuming a power of 2 dimension for the original picture, the split is performed as follows. Starting with the original image, its L^2 approximation is evaluated with a set of 2-D approximating functions. Whenever the values of the quality indices are beyond their respective acceptance threshold, the initial square is split into four squares of identical size. The same procedure is iterated for every subsquare until the quality measure becomes satisfactory. After the segmentation procedure, the 2-D signal is represented by the location of the different segmented regions and 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 [16].

The merge process is used to associate different regions obtained by the split operation in order to obtain a more efficient segmentation of the original image. Any segmentation algorithm requires the definition of an appropriate data structure in order to effectively access and relate the different regions. The data structure chosen to merge various squares obtained by the split algorithm is the Region Adjacency Graph (RAG). 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 regions (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 regions (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. An example of this operation is presented in Fig. 5. These images are segmented into 49 regions with compression ratios ranging from 42: 1 to 68: 1. Two termination criteria were considered to stop this part of the segmentation: 1) the minimum number of regions of the segmented image and 2) the maximum acceptable dissimilarity between the original image and the approximated one.

As post processing, a smoothing technique can be used to enhance the quality of the segmented image in case of polynomial approximation. This procedure can be used after the split-and-merge. Due to the structure of polynomial functions, the approximated signal between adjacent regions may be discontinuous. 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. 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.

IMAGE SEQUENCE CODING BY SPLIT AND MERGE

Ideally, in object-based image sequence coding one should first determine all the objects in the first frame of the sequence. Then, events like motion, enlargement, modification and the disappearance or appearance of new objects need to be detected and analyzed. There are several techniques that could be employed to try to accomplish these tasks. Segmentation has been successfully used in static object-based coding. It can be obtained in many ways: contour extraction, split and merge, region growing, etc. All these methods do not, of course, give the same results, but they all attempt to extract regions whose frontiers match the borders of the objects in the scene. On the basis of the high performance obtained in the static case, split and merge is applied to image sequences in our current work. Past experience suggests that the coding method be matched to the nature of the data, i.e. to its 3-D character. Therefore, a 3-D split-and-merge algorithm is investigated for low bit rate image sequence coding.

The data we are aiming at compressing result from a digital image sequence, where each image is 256x256 pixels in size, and the images are transmitted at 25 images/sec. There is thus a tridimensional data space (*x,y* space + time). In the method to be described below, the sequence is divided or segmented into various regions. A region is a set of contiguous volume elements, or voxels, which share one or several properties. In our current work, we define a region as a regular domain in the image sequence, over which the grey level variation can be approximated within a specified error, by a 3-D polynomial. A region can thus be represented by the coefficients of its approximating polynomial and by the description of its 3-D border. To find these regions, we use a 3-D split and merge algorithm, which consists of starting from an initial region space and then merging these regions according to the properties they share.

To get the initial region space, from which the final regions will be grown, the 3-D data space can be split into regions such that the luminance in each region is closely approximated by a 3-D polynomial. The initial regions are obtained in the following way: First, we consider the entire image sequence, trying to approximate it by a single polynomial. As the image sequence does not usually have a homogeneous luminance, its approximation by one polynomial is not usually acceptable, because the resulting approximation error is very high. In that case, the entire data space is split in the three directions (*x,y* and time) to get 8 smaller volumes. Then, the approximation is performed on each one of those volumes. If the approximation error is too high for a given volume, the volume is split again, and so on. The process stops when the approximating error on each sub-volume is lower than a predefined value. Finally, at the stop level, each sub-volume will be considered as an initial region.

Once the initial region space has been obtained, the most similar regions must be merged together. For that, a region adjacency graph is used as before. As shown in Fig.6, a region adjacency graph (RAG) is a graph in which each node represents a region and each branch represents an interconnecting link between two neighboring regions. A cost is assigned to each branch of the RAG to indicate the similarity of neighboring regions. The more similar the regions are, the lower the cost is. Merging the regions sharing similar properties is performed with a priority order. The regions whose interconnecting cost is minimum are merged first. In this way, we assure that the growth will be homogeneous and isotropic. By iteratively repeating the merge of the most similar regions, we reduce the redundant information contained in the various region attributes. The merge process stops when no more regions should be merged, either because the interconnecting cost is too high, indicating high dissimilarity of the regions, or because a predefined minimum of regions has been reached.

In our application, the grey level variation over a region is approximated by a polynomial function. The interconnecting error of neighboring regions R_i and R_j is the error that results from the approximation of the image over the joint region $R_i » R_j$. We have chosen to calculate the approximating error in the least mean squared sense. As regions can have any size, any shape and any grey level variation, the approximating error can, in general, take a large range of different values, from less than 10^{-3} to more than 10^{8}. To have a smaller range of the possible values for the interconnecting cost, the cost will be defined as a logarithmic function of the approximating error.

The function which has been chosen to approximate the image over each 3-D domain is a polynomial of first degree in x and y and of degree 0 in t. This choice can be justified because the grey level variation in the time direction is often not very significant between two successive frames. The first degree in x and y allows the representation of linear grey level variations over the object. Thus, the interpolation polynomial Q is given by $Q(x,y,t) = c + ax + by$, where a, b and c are the coefficients which must be determined in order to have the best approximation of the image for a given region.

To code the borders of the regions, a pyramidal structure is used, composed of a set of parallelepipeds of various sizes. The domain of each region can be thus represented by a set of parallelepipeds of appropriate dimensions (fig. 7). To represent the border of a region as well as possible, the set of the parallelepipeds must match exactly the region that it defines. By setting some constraints on the dimensions and on the positions of the parallelepipeds in the pyramidal structure, it is possible to get a compact code to describe the borders of all regions.

To obtain the most compact pyramidal structure representation, the pyramid is obtained by splitting the data volume along three different directions: x=constant, y=constant and time t=constant. As each cut performed on a volume give rise to two new sub-volumes, the set of the cuts which are performed on the sequence can be represented using a binary tree, where each branch in the binary tree indicates a subdivision in the x,y or time direction.

To get the shapes of the regions, we have to determine which parallelepiped belongs to which region. For that, a label is assigned to each region, such that two neighboring regions do not have the same label. The parallelepipeds will be labeled according to the region to which they belong. Thus, two neighboring parallelepipeds will have the same label if, and only if, they belong to the same region. It has been shown that 8 different labels are enough to properly label a tridimensional graph. However, although it is theoretically possible to get this minimum of 8 colors, the computer time that would be spent to obtain a maximum of 8 colors could be extremely large. The labeling method we have used does not attempt to reduce the number of the colors to its absolute minimum, but it has the advantage of being very fast. Despite the fact that we do not seek an absolute minimum, the number of colors used is still quite small and has never exceeded 13. Thus, 3 to 3.5 bits in average will be enough to code the label of a parallelepiped.

The coefficients of the approximating polynomials are coded in a straightforward manner, using 16 bits: 5 bits are used for coefficient a and b, and 6 bits are used for the c coefficient. The c coefficient, representing the average grey level, is judged more important than the coefficients a and b, representing the grey level variation along the x and y coordinates. For this reason, the c coefficient has been coded with more precision than the other coefficients.

Tests of the proposed algorithm have been performed using standard image sequences. Image sequences 256x256 pixels in size, transmitted at 25 images/sec have been compressed by a factor of approximately 200 to 300, thus allowing transmission through a 48 to 64 kbits/sec channel. The quality of the restored images is reasonably good. However, additional work is required to remove some side effects of the split and merge and reconstruction processes. The main quality defects are the artifacts located at the border of two regions which have been formed by the merge of large initial regions (large cubes). Another defect in the image quality is the imprecision of the borders of the objects in a scene. This is principally due to the size of the smallest possible initial region, which has been fixed at 2x2x2 voxels, to avoid indetermination, while performing approximating error calculation, in the least-mean-squared sense.

CONCLUSIONS

In this paper a brief overview is given of image coding techniques using the contour-texture model. The compression ratio achievable with these techniques may reach very high values. In contrast with conventional methods using a signal processing approach in the selection of the messages to be coded, they are based on scene analysis features. These new methods put heavy emphasis on the selection of the messages to be coded. In this context, signal processing approaches are not as successful as pattern recognition or artificial intelligence approaches. The coding is done in the classical way. It is clear that the methods we have presented need several improvements to produce better quality images at the same compression ratio or to reach higher compression ratios for the same quality. More detailed results can be found in [17].

Because of computer memory limitations, the split and merge algorithm has not been tested on sequences longer than 32 frames. By performing the segmentation algorithm on a longer sequence (a few seconds in length), the compression ratio could be considerably increased, because of the redundant information contained in successive frames. This remark is valid only if the movements in the scene are not too large, and if the border on the time axis is constrained to the same scene. The compression ratio could also be increased by improving the coding of the polynomial coefficients, by using vector quantization coding rather than fixed length coding.

The goal to be reached is to segment the image into regions corresponding to the real objects of the scene, without missing small ones and without introducing false objects and hence false contours. Powerful representations should be designed to describe the grey level evolution within each region. Recent efforts in texture analysis and synthesis will be of great value to image coding to render the natural look when added to the representation of regions. It is hoped that image coding will remain a center of interest for researchers and that even higher compressions will be obtained.

ACKNOWLEDGMENTS

The author expresses his thanks to his colleague Dr. T. Reed and to his Ph.D. student Mr. P. Willemin for their help in the preparation of the manuscript.

REFERENCES

[1] M. Kunt, A. Ikonomopoulos and M. Kocher, "Second Generation Image Coding Techniques" (Invited Paper), Proc. IEEE, Vol. 73, No. 4, April 1985, pp. 549-574.
[2] M. Kunt, "Edge Detection : A Tutorial Review" Proc. ICASSP 82, Paris, May 2-5, 1982, pp 1172-1176.
[3] A. Ikonomopoulos, M. Kocher, and M. Kunt, "Image Coding Based on Human Visual System Propertyies for Optimal Reduction of Redundancy", Proc. 3rd Scandinavian Conference on Image Analysis, Copenhagen, Danemark, July 12-14, 1983, pp. 216-222.
[4] M. Kunt, A. Ikonomopoulos and M. Kocher (invited lecture) "Compression d'images : Methodes de la deuxie`me generation" Premier Colloque GRETSI-CESTA, Biarritz, France, May 21-25, 1983.
[5] A. Ikonomopoulos and M. Kunt, "High Compression Image Coding Via Directional Filtering" Signal Processing, Vol. 8, No. 2, April 1985, pp. 179-203.
[6] D.H. Hubel and T.N. Wiesel, "Brain Mechanism of Vision", Sci. Amer., Vol. 241, pp.150-162, Sept. 1979.

[7] M. Benard and M. Kunt, "Linear Prediction in Directional Images" Proc. EUSIPCO-86, 2-5 Sept. 1986,The Hague, The Netherlands, North Holland.
[8] M. Kocher and M. Kunt," A Contour-texture Approach to Picture Coding", Proc. ICASSP-82, Paris, May 1982, pp. 436-440.
[9] M. Kocher,"Codage d'images a` haute Compression base sur un modèle Contour-texture" Ph.D. Thesis, No. 476, Dept. of Electrical Engineering, Swiss Federal Institute of Technology, Lausanne, Switzerland, March 1983.
[10] M. Kocher and M. Kunt, "Image Data Compression by Contour-texture Modelling", SPIE Int. Conference on the Applications of Digital Image Processing, Geneva, April 1983, pp. 131-139.
[11] M. Kocher and M. Kunt, "A Contour-Texture Approach to Picture Coding" Proc. Melecon-83, Athens, Gre`ce, May 24-26, 1983, Paper C2.03.
[12] M. Eden and M. Kocher, "On the Performance of a Contour Coding Algorithm in the Context of Image Coding. Part II : Coding and Contour Graphs" Signal Processing, Vol. 8, May 1985, to be published.
[13] R. Leonardi, "Segmentation adaptative pour le codage d'images", Ph.D. Dissertation, No. 691, Dept. of Electrical Engineering, EPFL, July 1987.
[14] M. Kocher and R. Leonardi, "Adaptive Region Growing Technique Using Polynomial Functions for Image Approximation", Signal Processing, Vol. 11, No. 1, July 1986.
[15] R. Leonardi and M. Kunt, "Adaptive split and merge for image analysis and coding" SPIE Int. Symp. Image Coding, Cannes, France, Dec. 2-6, 1985.
[16] 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.
[17] M. Kunt, M. Benard and R. Leonardi, "Recent Results in High Compression Image Coding" (invited paper), IEEE Trans. on Circuits and Systems, Vol. CAS-34, No. 11, November 1987.

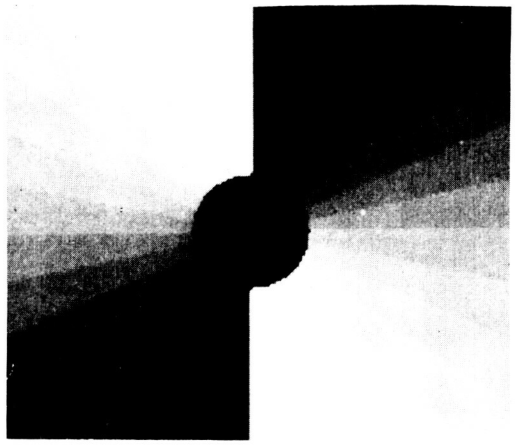
Fig. 1. Sectors of the Fourier domain covered by directional filters.

(a)

(b)

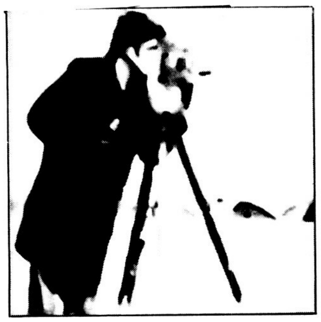

Fig. 2. Directional decomposition based coding results. The compression ratios are 57:1, 59:1 and 49:1 respectively.

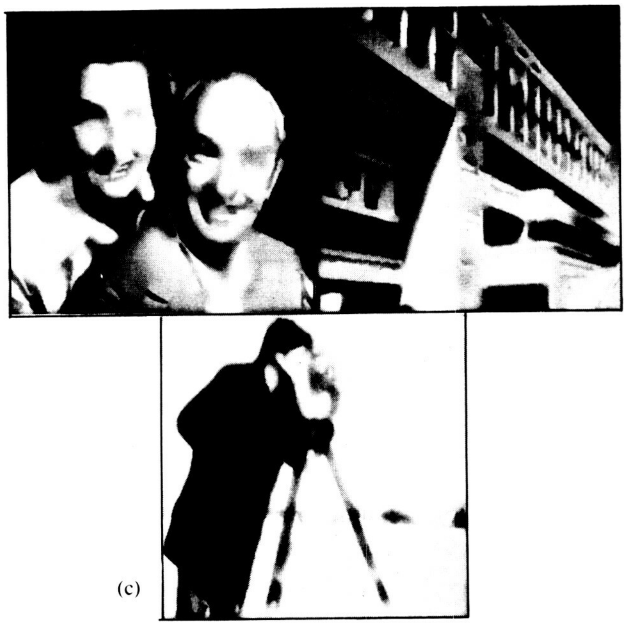

Fig. 3. Directional decomposition based coding results. The compression ratios are 118:1, 86:1 and 84:1 respectively.

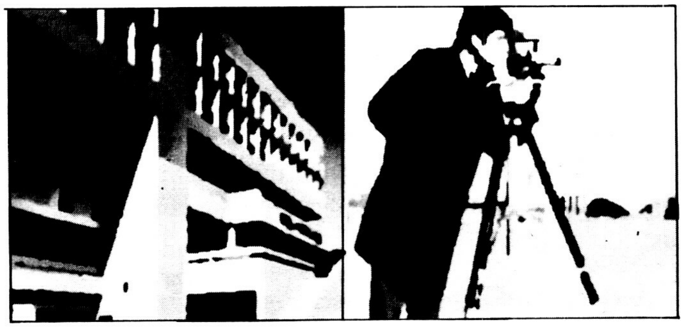

Fig. 4. Region growing based coding results. Compression ratios are 44:1, 38:1 respectively.

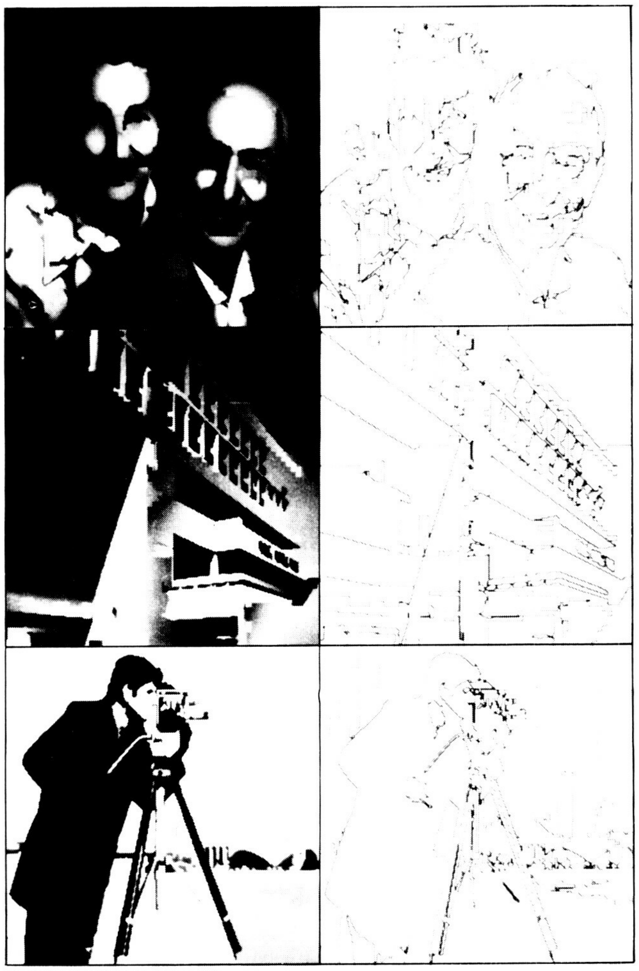

Fig. 5. Result of merge process. Each image is segmented into 49 regions. The compression ratios are 42:1 with third order polynomials (a), 53:1 with first order polynomials (b) and 68:1 with zero order polynomials.

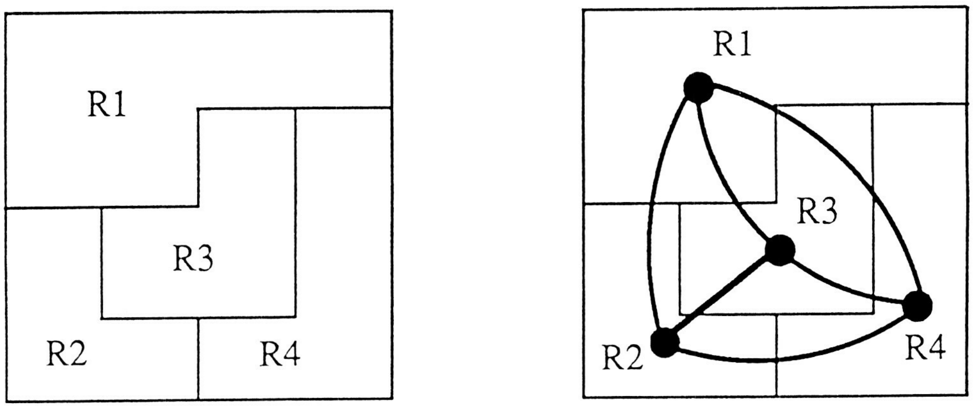

Fig. 6. Region space and its corresponding region adjacency graph (bidimensional representation)

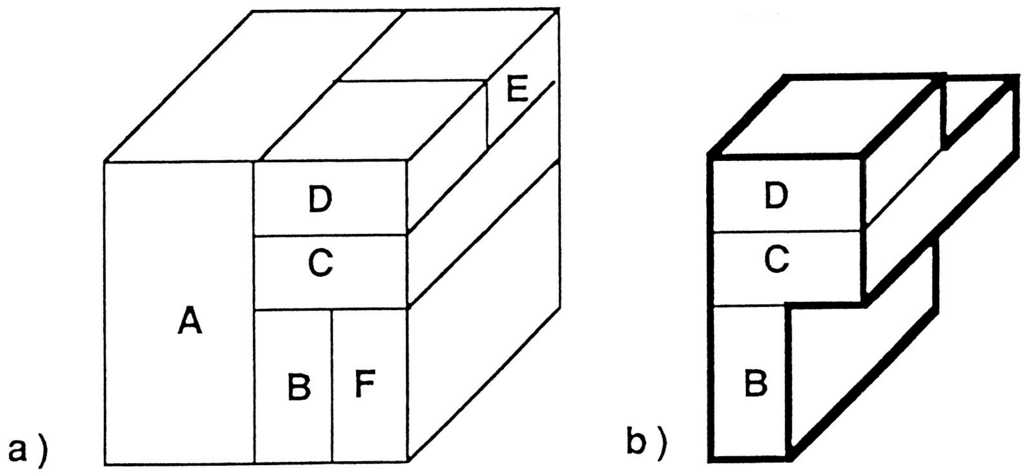

Fig. 7. a) Pyramidal structure defined in the entire data space.b) Region built from a set of parallelepipeds belonging to the pyramidal structure.