SUCCESSIVE INTERPOLATION/EXTRAPOLATION SCHEME FOR MISSING SEGMENTS RESTORATION

Anton BŘEZINA, Jaroslav POLEC, Andrej VARCHOLA
Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia, E-mail: brezina@ktl.elf.stuba.sk

SUMMARY

This paper introduces a method for restoration of missing arbitrarily shaped image segments using combined discrete orthogonal transforms (DOT). The missing segment texture is successively interpolated/extrapolated using the texture of the correctly decoded segments belonging to the same object, thus having the same texture pattern.

Keywords: *discrete orthogonal transform, image interpolation/extrapolation, image coding, triangulation*

1. INTRODUCTION

Lately, great concern in image processing is devoted to region-oriented methods. Region-oriented image representation offers several advantages over block-oriented approach, e.g. adaptation to the local image characteristics. New algorithms are necessary for image coding, if we work with arbitrarily shaped image regions, called segments, instead on rectangular blocks. The original approach for the coding of arbitrarily shaped image segments based on a generalized orthogonal transform was discussed in [1]. Application scheme with cosine transform is proposed in [2]. The coding method used in this paper was thoroughly described in [3].

The transmission of images coded by block or segment based techniques via a noisy channel may lead to block or segment loss. Therefore error detection and concealment at the decoder side has to be applied. Commonly, spatial error concealment is used. It utilizes the surrounding correctly received image information to restore the damaged or missing pixels.

A standard approach [4] assumes that the image content is changing smoothly. Hence the algorithm tries to restore the transition across the block boundary as smooth as possible. The extrapolation-based method of [5] tries to reconstruct the missing pixels as a weighted linear combination of correctly received pixels. Hence, the method can lead to solving a great number of linear equations and is computationally very complex. The transmission of block-coded image data in via wireless channel is described in [6]. Very interesting and novel method for spatial error concealment based on successive extrapolation of missing blocks is described in [7].

2. SUCCESSIVE INTERPOLATION/EXTRAPOLATION

In [3] a method for hybrid approximation of non-square areas was developed. The texture of an area is successively approximated and then cut to the shape of the segment. This principle is used in this paper for error concealment by estimating the missing image content using the surrounding area belonging to the same object, thus having the same texture pattern. The image content of the known segments is successively approximated using combined DOTs and the missing segment is obtained by interpolation/extrapolation (see Fig. 1).

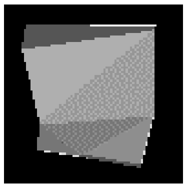


Fig. 1 Triangular image segments belonging to the same object. Texture pattern inside each of them can be obtained via interpolation/extrapolation from the texture pattern of the others.

2.1. Image segmentation

This approach to image partitioning was based on unsupervised segmentation method for colour – texture regions [8].

This method does not attempt to estimate the specific model for a texture region. Instead, it tests for the homogeneity of a given colour-texture pattern, which is computationally more feasible than model parameter estimation. In order to identify this homogeneity, the following assumptions about the image are made:

- Each image contains a set of approximately homogeneous color-texture regions. This assumption conforms to our segmentation objective.
- The color information in each image region can be represented by a set of few quantized colors.
- The colors between two neighbouring regions are distinguishable – it is a basic assumption in any color image segmentation algorithm.

The segments are found in few steps. First, colours in the image are quantized to several representative classes and then their corresponding colour class labels replace the pixel values. In this way we obtain a class-map. The class-map can be viewed as a special kind of texture composition. In the class-map so called *J*-values are solved from local neighbourhood of a pixel. These *J*-values correspond to the minimum variation of texture in image regions. The measure *J* is defined as

$$J = \frac{SB}{SW} = \frac{ST - SW}{SW} \qquad (1)$$

where *SW* is the sum of all so called within-class variances of points belonging to distinct colour classes in the image and *ST* is the total variance of image points in the class map.

The criterion measures the distances between different classes *SB* over the distances between the members within each class *SW.* A higher value of *J* indicates that the classes are more separated from each other and the members within each class are closer to each other, and vice versa.

For the case when an image consists of several homogeneous color regions, the color classes are more separated from each other and the value of *J* is large. On the other hand, if all color classes are uniformly distributed over the entire image, the value of *J* tends to be small. The larger the *J*-value is, the more likely the corresponding pixel lies near a region boundary.

```
+ + + + + o o o o     + * + * + * + * +     + + + + + * o * o
+ + + + + o o o o     o + o + o + o + o     + + + + + o * o *
+ + + + + o o o o     + * + * + * + * +     + + + + + * o * o
+ + + + + o o o o     o + o + o + o + o     + + + + + o * o *
+ + + + + o o o o     + * + * + * + * +     + + + + + * o * o
+ + + + * * * * *     o + o + o + o + o     + + + + * o * o *
+ + + + * * * * *     + * + * + * + * +     + + + + o * o * o
+ + + + * * * * *     o + o + o + o + o     + + + + * o * o *
+ + + + * * * * *     + * + * + * + * +     + + + + o * o * o
        a)                    b)                    c)
```

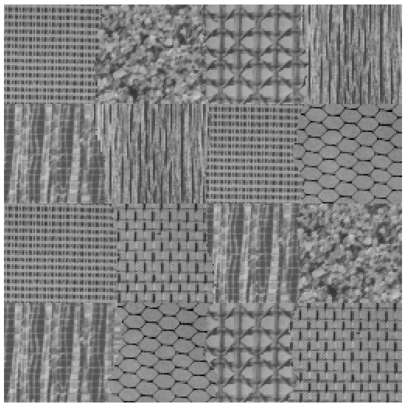
Fig. 2 An example of *J*-values for different types of texture maps: a) *J*=1.72, b) *J*=0, c) *J*=0.855

Examples of a class-map are shown in Fig. 2, where label values are represented by three symbols, ‘*’, ‘+’, and ‘o’. Usually, each image region contains pixels from a small subset of the color classes and each class is distributed in a few image regions.

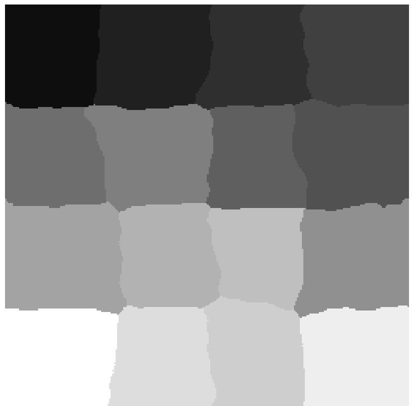
Finally, a region growing and merging method is applied to the image of *J*-values, so called *J*-image, to obtain the final segmentation. *J*-images correspond to measurements of local homogeneities at different scales, which can indicate potential boundary locations. A spatial segmentation algorithm grows regions from seed areas of the *J*-images to achieve the final segmentation.

The *J*-image allows us to use a multiscale region-growing method. Consider the original image as one initial region. The algorithm starts to segment all the regions in the image at an initial large scale. It then repeats the same process on the newly segmented regions at the next smaller scale until the minimum specified scale is reached.

The over-segmented regions after region growing are merged based on their color similarity. The pair of regions with the minimum distance is merged together. The process continues until a maximum threshold for the distance is reached.



a)



b)

Fig. 3 The original image a) and the segmentation map obtained using JSEG method b)

2.2. Polygonal approximation of the region boundaries

The polygonal approximation of the regions obtained in the segmentation phase is vital for the simplicity of segment borders coding. As the segments can be arbitrarily shaped, we have to remove very small areas and simplify the borders, so they can be described as a sum of simple lines. This also simplifies the division of segments into non-ovelapping triangles as we will se later.

First, the boundaries of all input regions are to be found. The boundary point is each point, its gray value is equal to region gray value, but the gray value of at least one of neighboring pixels differs from the region gray value. To find the boundaries, 8-directional algorithm based on LML (left-most-looking) rule was used [9]. Now each segment is described by own boundary, so that between neighboring segments there are in all cases two paralel boundaries, one belonging to each segment.

These “doubled” boundaries between the segments are reduced to be only one pixel wide, so that the boundary between two adjacent segments is coded only once. Also very small segments are being attached to their larger neighbours in this step. This has also significant impact on the resulting code efficiency.

The boundaries of the segments are simply approximated with polygons.

It has been shown [10] that the boundary degradation caused by this step has only a small influence on the quality of the resulting image, but is very important for increasing the code efficiency.

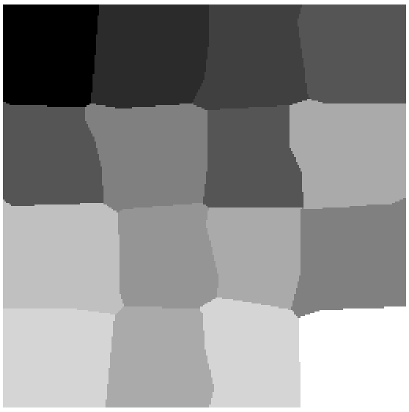


Fig. 4 Segmentation map from Fig. 3 after polygonisation of the region boundaries

2.3. Triangulation of the image objects

In the segmentation process we have obtained regions, which correspond to real objects of the scene based on the segmentation criterion. However, these objects may be too large, to be processed directly, moreover, we need more detailed segmentation for the purpose of successive interpolation/extrapolation. In this phase the objects are divided into set of non-overlapping triangles. The triangle was chosen to be the most suitable shape to approximate the image segments. As the boundaries have already been polygonised, the set of triangles should perfectly match each segment shape.

We have used constrained Delaunay triangulation in our work.

2.3.1. Constrained Delaunay triangulation

A Delaunay triangulation of a point set is a triangulation, whose vertices are the point set, having the property that no point in the point set falls in the interior of the circumscribing circle (circle that passes through all three vertices) of any triangle in the triangulation [11]. A Planar Straight Line Graph (PSLG) is a collection of points and segments. Segments are simply constrained edges, whose endpoints are points in the PSLG. A constrained Delaunay triangulation of a PSLG is similar to a Delaunay triangulation, but each PSLG segment is present as a single edge in the triangulation. A constrained Delaunay triangulation is not truly a Delaunay triangulation. Effective implementation of Delaunay triangulation usually needs O(N•logN) time, for some cases O(N) time can be achieved. Constrained edges in (CDT) cause the increase of computational time. This increase depends on the number of constrained edges. For our purpose, we use A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator (software called Triangle) by J. R. Shewchuk [11].

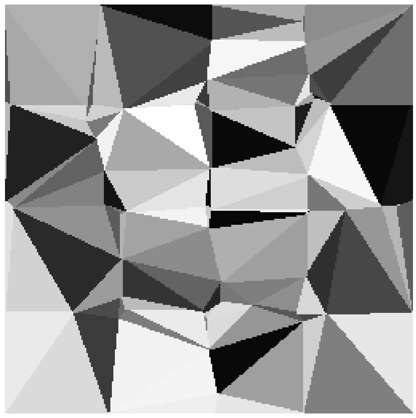


Fig. 5 Triangular mesh obtained using [11] on polygonised segmentation map from Fig. 4

2.4. Selection of suitable basis function

Let us consider, that we have decoded an image coded using [3]. If some of the blocks have been decoded incorrectly, or were damaged and we possess the knowledge of the location of objects, that have the same texture pattern, we can interpolate/extrapolate the missing segment from the correctly received segments that belong to the same object.

Let us have a segment *A* of a 2D discrete image. This segment represents an arbitrarily shaped region with internal texture structure $f(n_1, n_2)$. First we find circumscribing rectangle *L* of this segment. Next, we pad this rectangle with zeros so that its width and height will be the power of 2 thus allowing us to use the fast transform algorithms. The size of the rectangle is now $N_1 \times N_2$ points. The texture structure of segment *A* in rectangle *L* is then approximated using the appropriate basis functions.

In 1D case, if $x(n) = \{x(0), x(1), ..., x(N-1)\}$ is the set of points representing the image grey level values and $u_k(n)$ is the set of orthogonal basis functions, we can obtain the belonging spectral coefficients as [12]

$$X(k) = \sum_{n=0}^{N-1} x(n).u_k^*(n), \ k = 0,1,..., N-1 \tag{2}$$

In each step of approximation, the suitable basis function is selected until we approximate the texture with sufficient quality. This method is called the “Matching Pursuit” algorithm [13]. According to more recent work [14] we provide an overcomplete dictionary of basis functions for the approximation process which is constructed as a mixture of basis functions of more DOTs.

We have a set of orthogonal basis functions defined as

$$u_{k_1,k_2}(n_1,n_2),\ k_1 = 0,1,...N_1-1,\ k_2 = 0,1,...N_2-1,$$

where $n_1 = 0,1,...N_1-1$, $n_2 = 0,1,...N_2-1$.

After ν iteration steps, we have approximation $g^{(\nu)}(n_1,n_2)$ of segment texture $f(n_1,n_2)$. Using linear approximation theory, this can be expressed as sum of basis functions weighted by appropriate spectral coefficients.

$$g^{(\nu)}(n_1,n_2) = \sum_{k_1,k_2\in K_\nu} c^{(\nu)}_{k_1,k_2} \cdot u_{k_1,k_2}(n_1,n_2)\,, \qquad (3)$$

where K_ν denotes the set of basis function indices used in $g^{(\nu)}(n_1,n_2)$ and $c^{(\nu)}_{k_1,k_2}$ the spectral coefficients. The residing difference between the original texture and its approximation is then

$$r^{(\nu)}(n_1,n_2) = f(n_1,n_2) - g^{(\nu)}(n_1,n_2) \qquad (4)$$

Now we want to approximate this difference with a suitable basis function to minimize the error function

$$E_A = \sum_{n_1,n_2\in A} [f(n_1,n_2) - g(n_1,n_2)]^2\,, \qquad (5)$$

where A stands for the approximated segment. According to [3], we are searching the basis function that maximizes

$$\Delta E_A^{(\nu)} = \frac{\left[\sum_{n_1,n_2\in A} r^{(\nu)}(n_1,n_2)\cdot u_{k_1,k_2}(n_1,n_2)\right]^2}{\sum_{n_1,n_2\in A}\left[u^2{}_{k_1,k_2}(n_1,n_2)\right]} \qquad (6)$$

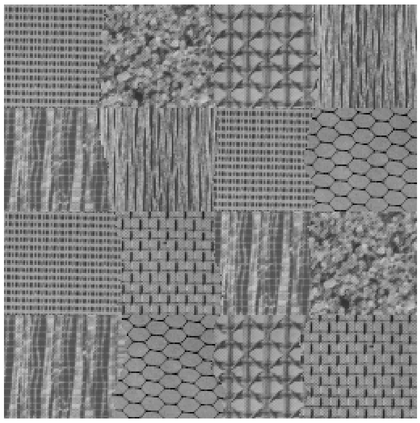
Because we use a dictionary constructed out of more DOTs, which results in an overcomplete set of basis functions, it is necessary that all of the basis function are normalized, so that none of them is preferred in the approximation process. More background information about signal decomposition into overcomplete systems can be found in [13], [14].

2.5. Missing segment interpolation/extrapolation

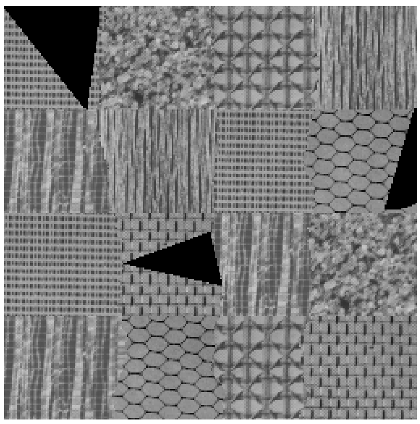
After the correctly received segments of an object have been approximated with sufficient quality (stated as criterion on marginal value of PSNR or MSE), the missing segment is interpolated/extrapolated as the appropriate part of the rectangle L circumscribing the whole object.

3. EXPERIMENTAL RESULTS

We treat each triangular segment as a separate object belonging to one arbitrarily shaped image segment. The texture inside each triangle is approximated and the resulting spectral data is considered to form a paket inside the whole bit stream. We then suppose that one of these pakets is damaged during the transmission and we need to restore it in the receiver.



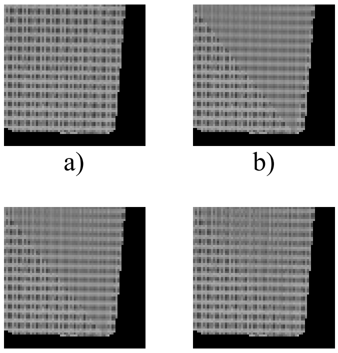
a)



b)

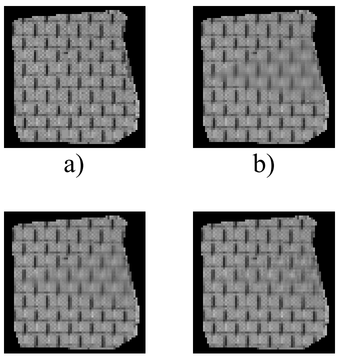
Fig. 6 a) Encoded image using [3], PSNR=45dB, b) damaged image

If in the segmentation process real objects of the scene have been separated, only the texture synthesis is needed, as the objects are supposed to be consisting of a uniform texture. This is done using the successive interpolation/extrapolation as described in this paper. The discrete cosine transform II (DCT) and Walsh-Hadamard transform (WHT) were used in the implementation.



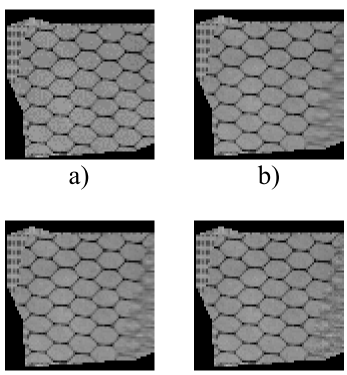
a) b) c) d)

Fig. 7 Detail on reconstructed segment
a) correctly received segment, b) reconstructed using 15 DCT and 2 WHT coefficients, c) reconstructed using 37 DCT and 7 WHT coefficients, d) reconstructed using 419 DCT and 318 WHT coefficients



a) b) c) d)

Fig. 8 Detail on reconstructed segment
a) correctly received segment, b) reconstructed using 44 DCT and 4 WHT coefficients, c) reconstructed using 80 DCT and 19 WHT coefficients, d) reconstructed using 750 DCT and 519 WHT coefficients



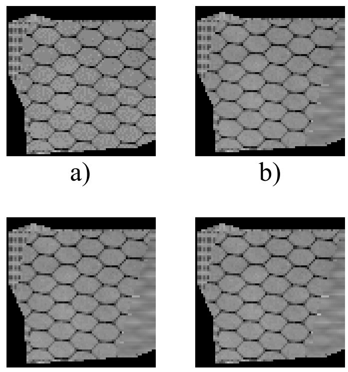
a) b) c) d)

Fig. 9 Detail on reconstructed segment
a) correctly received segment, b) reconstructed using 200 DCT and 42 WHT coefficients, c) reconstructed using 397 DCT and 158 WHT coefficients, d) reconstructed using 1727 DCT and 1073 WHT coefficients

4. CONCLUSION

This paper describes a missing segment restoration method using DOTs. Although several approximation algorithms using wavelet transforms were proposed [15], for extrapolational purposes only globally adaptive wavelet transforms can be used. The use of locally adaptive wavelets can lead to a large number of spectral coefficients used in the reconstruction process without any visual improvements.

The discrete Haar transform (DHT) was used for the demonstration.



a) b) c) d)

Fig. 10 Detail on reconstructed segment
a) correctly received segment, b) reconstructed using 23 DCT and 189 DHT coefficients, c) reconstructed using 27 DCT and 346 DHT coefficients, d) reconstructed using 41 DCT and 1490 DHT coefficients

REFERENCES

[1] M. Gilge, T. Engelhart, R. Mehlan, "Coding of Arbitrary Shaped Image Segments Based on a Generalized Orthogonal Transform", *Signal Processing: Image Communication*, 1, 1989, pp. 153-180.

[2] A. Kaup, T. Aach, "Coding of Segmented Images Using Shape-Independent Basis Functions", *IEEE Transactions on Image Processing*, vol. 7, 1998, no. 7, pp. 937-947.

[3] T. Karlubíková, J. Polec, A. Březina, "Hybrid Orthogonal Approximation of Non – Square Areas", *Proceedings of ICCVG 2004*. Warszawa, 2004, CD ROM, pp. 6.

[4] Y. Wang, Q.-F. Zhu and L. W. Shaw, "Maximally smooth image recovery in transform coding", *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1544–1551, Oct. 1993.

[5] T. Karlubíková, J. Polec, "Discrete Orthogonal Transform for Gappy Image Extrapolation", *Proceedings of ICCVG 2004*. Warszawa, 2004, CD ROM, pp. 6.

[6] S. D. Rane, G. Sapiro, M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and

compression applications”, *IEEE Transactions on Image Processing*, vol. 12, 2003, no. 3, pp. 296-303.

[7] K. Meisinger, A. Kaup, “Spatial error concealment of corrupted image data using frequency selective extrapolation”, *Conf. Rec. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, May 2004.

[8] Y. Deng, B., S. Manjunath, “Unsupervised Segmentation Method for Colour – Texture Regions in Images and Video”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI ’01)*, August 2001, http://vision.ece.ucsb.edu/publications/01PAMIJseg.pdf.

[9] R., C. Gonzalez, P. Wintz, *Digital image processing,* Second ed., Reading. Addison-Wesley Publishing Company, Tokyo, 1987, 503 p.

[10] J. Polec et al., “New Scheme for region approximation and coding with shape independent transform”, *PHOTO-GRAMMETRIC COMPUTER VISION 2002*, Graz, 2002, 19 Posters of WG 8 "Reliability and Performance of Algorithms".

[11] J. R. Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator”, *First Workshop on Applied Computational Geometry*, Philadelphia, Pennsylvania, pp. 124-133, ACM, May 1996.

[12] Ph. W. Besslich, T. Lu, *Diskrete Orthogonal-transformationen,* Berlin: Springer-Verlag, 1990, 312 pp., ISBN 3-540-52151-8.

[13] S. Mallat, Z. Zhang, “Matching Pursuit in a time-frequency dictionary”, *IEEE Transactions on Signal Processing*, Vol. 41, 1993, No. 12, pp. 3397-3415.

[14] S. S. Chen, D. L. Donoho, M. A. Saunders, “Atomic decomposition by basis pursuit”, *SIAM Review*, Vol. 43, 2001, No. 1, pp. 129-159.

[15] R. Vargic, “Wavelet-based compression of segmented images”, *Proceedings EC-VIP-MC 2003*, Zagreb, Croatia, 2-5 July 2003, pp.347-351.

BIOGRAPHY

Anton Březina was born on 1980 in Topoľčany, Slovak Republic. In 2004 he graduated (MSc.) with distinction at the Department of Telecommunications of the Faculty of Electrical Engineering and Information Technology at Slovak University of Technology in Bratislava. Since 2004 he is studying as a PhD. student at the Department of Telecommunications. His scientific research is focusing on signal and image processing, image compression and reconstruction.

Jaroslav Polec was born on 1964 in Trstená, Slovak Republic. He received the Engineer and PhD. degrees in telecommunication engineering from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in 1987 and 1994, respectively. From 1997 he is associate professor at the Department of Telecommunications of the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology and from 1998 at the Department of Computer Graphics and Image Processing, Faculty of Mathematics, Physics and Informatics of Comenius University. He is director of the Telecommunication Users Group of Slovakia and member of the Group for Digital Broadcasting of the Slovak Republic. He is member of IEEE. His research interests include Automatic-Repeat-Request (ARQ), channel modeling, image coding, interpolation and filtering.

Andrej Varchola was born in 1980 in Prešov, Slovak Republic. In 2002 he received the Bc. degree in information technology from the Faculty of Electrical and Information Technology, Slovak University of Technology. Currently, he is the M.Sc. student of telecommunication engineering at the Slovak University of Technology. His research interests include image coding.