Computación y Sistemas Vol. 9 Núm. 3, pp. 256-269
© 2006, CIC-IPN, ISSN 1405-5546, Impreso en México

Spatial Error Concealment Based on Bezier Curves

Ocultamiento de Errores Espacial Mediante Curvas de Bezier

Rogelio Hasimoto-Beltrán[1] and Ashfaq A. Khokhar[2]
[1] Centro de Investigación en Matemáticas – CIMAT
Jalisco s/n, Col. Mineral de Valenciana, Guanajuato, Gto., México 36240
Tel: +52-(473)-732-7155 ext. 49636, Fax: +52-(473)-732-5749
hasimoto@cimat.mx
[2] University of Illinois at Chicago,
Departments of Computer Sciences and Electrical and Computer Engineering,
851 South Morgan Street, Chicago, IL 60607, USA
ashfaq@uic.edu

Article received on December 16, 2003; accepted on December 12, 2005

Abstract

We present a geometric error concealment scheme for Discrete Cosine Transform (DCT) based image/video data based on the Bezier Curves (BC). Our scheme unlike the original proposal of Zeng and Liu, 1999, makes use of the average edge direction and local curvature (extracted from healthy blocks around the damaged block) as boundary conditions to create an n-degree BC. The BC is then used to interpolate the edge of the lost block (8x8 pixels), as well as to directionally reconstruct its low frequency data. In those cases where more than one edge is found to cross the lost area, a cost function defined in terms of the local and global edge curvature is used to find their best match. Results show that our approach provides an almost perfect reconstruction and excellent subjective quality of the restored data, outperforming current linear interpolations schemes in the literature.
Keywords: Error Concealment, DCT-based compression, Bezier Curves.

Resumen

Se presenta un esquema nuevo para el ocultamiento de errores de transmisión (pérdidas de paquetes) en imágenes y video basado en las Curvas de Bezier (BC). El esquema propuesto a diferencia de la propuesta original de Zeng y Liu, 1999, hace uso de la dirección promedio de los bordes y curvatura local alrededor de la zona dañada como condiciones de frontera para crear una curva de Bezier de grado *n*. La BC es usada para interpolar los bordes del bloque perdido (8x8 pixels), así como también para la reconstrucción de la información correspondiente a las bajas frecuencias. En casos donde más de un borde atraviesa la zona dañada, se propone una función de costo en términos de sus curvaturas local y global para encontrar sus correspondientes pares y proceder con el proceso de interpolación (reconstrucción) de la información perdida. Los resultados muestran que nuestro esquema reconstruye casi perfectamente la información perdida (durante la transmisión) lo que repercute en una excelente calidad subjetiva de la imagen/video reconstruido. Nuestro esquema es muy superior a los interpoladores lineales encontrados en la literatura.
Palabras Clave: Ocultamiento de Errores, Compresión basada en la DCT, Curvas de Bezier.

1 Introduction

Error Concealment by post-processing (EC) is a widely used technique for data recovery in which the receiver takes full responsibility of the reconstruction process. It attempts to recover the lost data without relying on additional information from the source-end [Salama, et al., 2000]. EC does not guarantee a perfect recovery, but it can recover from errors when other techniques such as Forward Error Control (FEC) or data retransmission fail to do their job [Hasimoto, 2001]. Error concealment can be classified into two broad categories: ***temporal*** and ***spatial*** error concealment. In this work we concentrate on Spatial Error Concealment, where the corrupted area (lost block) is filled in by using information from adjacent error free blocks, rather than using information from previously decoded frames (as in temporal error concealment). In this category there are three main approaches found in the literature for recovering from transmission errors [Hasimoto, 2001; Pei-Chun, et al., 2002; Shahram, et al., 1999]: Minimum Distance Interpolation (MDI), Directional Interpolation (DI), and Bayesian Interpolation (BI). The first category makes use of the smoothness property among neighboring blocks, and the best reconstruction is achieved by an energy constrained minimization approach [Zhu, et al., 1993; Wang, et al., 1993; Hemami and Meng, 1995]. These methods in general, have a tendency to blur edges because the spatial variations are measured by calculating the difference between two adjacent pixels, which are in turn minimized by a cost function. DI schemes have been proposed to solve this problem. DI is based on the extraction of the local structure (e.g. direction of edges) around the missing block for its reconstruction [Kwok and Sun, 1993; Jung, et al., 1994]. These techniques have become very popular because of their simplicity and efficiency to recover low and high frequency components. Kwok and Sun, 1993, proposed a spatially correlated edge information scheme to perform a multidirectional interpolation (along 8 different paths) to restore the missing blocks. Jung, et.al, 1994, developed an algorithm similar to Kwok and Sun, 1993, except that they used the first layer of pixels surrounding the lost block to obtain the average direction for the interpolation process. Zeng and Liu, 1999, proposed a directional interpolation based on the geometric structure around the lost block as a viable alternative for a more pleasant reconstruction. In this approach, the direction of every edge transition is quantified and inserted in a cost function for the edge matching process. The use of a priori model along with the smoothness assumption among neighboring blocks is considered in the BI schemes. Markov Random Fields (MRF) has been used to model the image properties, because of its tractability and the ability of the model to capture non-Gaussian aspects of the image such as edges [Shahram, 1999]. Even though BI uses a priori model, not necessarily produces better results than the DI schemes.

In general, the above techniques give relatively good results under simple edge patterns, such as straight line crossing throughout the lost block. In this work, we propose a generalized interpolation scheme based on Zeng and Liu's scheme, capable of representing more complicated edge geometries (such as curved edges, rounded corners, etc.) by considering the average edge direction, local and global edge curvature and adaptive Bezier curves to model the geometric structure of the lost blocks. Figure 1 describes the entire scheme. The surrounding 4 layers of the lost block are first converted into a binary pattern by thresholding. The structure of the edge is inferred based on the position of the control points and curvature values at each transition point. Finally, directional interpolation along the curve is applied to the missing pixels in each region. A rough difference between our scheme and the one presented by Zeng and Liu is shown in Figure 2. Figure 3 shows the limitation of Zeng and Liu's scheme when a lost block is surrounded by curved edges. We will show that our proposed scheme produce high quality results, which are visually undistinguishable from the original information.

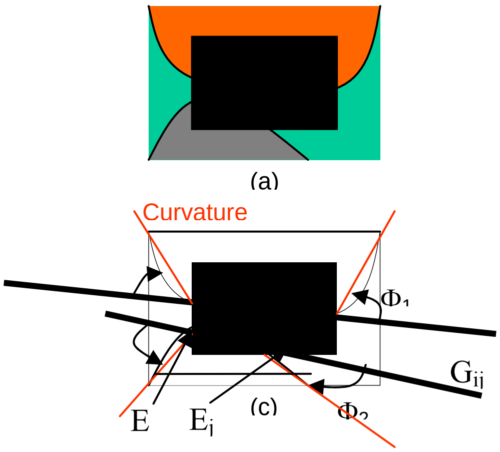


Fig. 1. General diagram of the proposed scheme. (a) The lost block; (b) binary pattern; c) inferred structure; and (d) reconstructed block through Bezier interpolation.

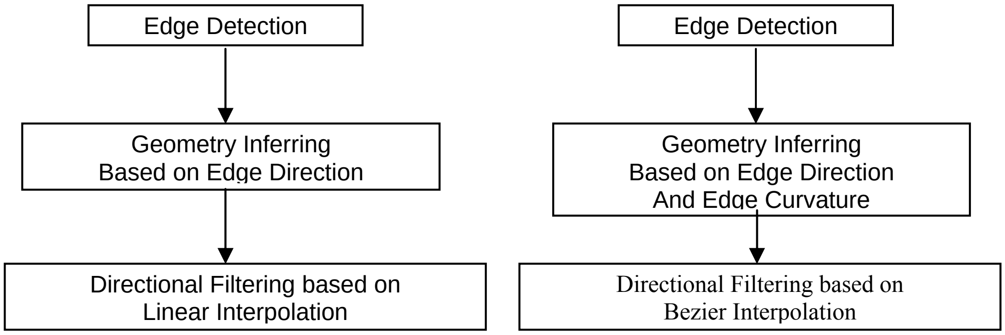

Edge Detection
Geometry Inferring Based on Edge Direction
Directional Filtering based on Linear Interpolation
Edge Detection
Geometry Inferring Based on Edge Direction And Edge Curvature
Directional Filtering based on Bezier Interpolation
(a)
(b)

Fig. 2. (a) Zeng and Liu's EC scheme, (b) our proposed scheme.

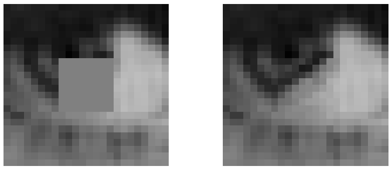


Fig. 3. Lost block (left) and its reconstruction based on Zeng and Liu, 1999 (right).

This paper is organized as follows. Section 2 provides a theoretical overview of BC; section 3 describes the proposed model, and presents a simplified scheme for the computation of the local edge curvature. Section 4 demonstrates the effectiveness of the model for restoring missing information with 1, 2, and 3 edge patterns (representing 2, 4, and 6 edge transition points respectively). Finally, conclusions are presented in section 6.

2 Bezier Curves (BC)

Bézier curves different than other curve modeling techniques (splines curves) are not constrained to pass through all the specified points, instead they only approximate the given points (called control points), as shown in Figure 4 [Farin, 1993]. Once the control points are given, the curve shape is determined. This behavior of the BC is an important characteristic useful to transform them into a powerful error concealment scheme. When a block is lost, the average features such as edge position and average tangent of the surrounding blocks can be entered into the BC model as control points (edge curvature is used for the edge matching process). The model will then find the best fitting curve under the current boundary conditions. Another characteristic of the BC is that it is not needed to specify additional dependencies of the control points in the lost block, as in the case of the spline curve model, the dependencies are already fixed by the model. Therefore, the key point consists in defining the best properties representing the lost block, in order to minimize the difference between the restored block and the neighboring blocks. For this to happen additional constrains and assumptions must be set forth as explained later on this section. The Bezier curves can be defined by a parametric function of the following form:

$$BC(u) = \sum_{i=0}^{n} P_i B_i^n (u)$$

where the vectors P_i represent the $n+1$ vertices or control points of a characteristic polygon (Figure 5), B_i^n are the Berstein Polynomials for $u \in [0,1]$, defined as:

$$B_i^n (u) = C(n,i) u^i (1-u)^{n-i}$$

and where C(n,i) is the familiar binomial coefficient:

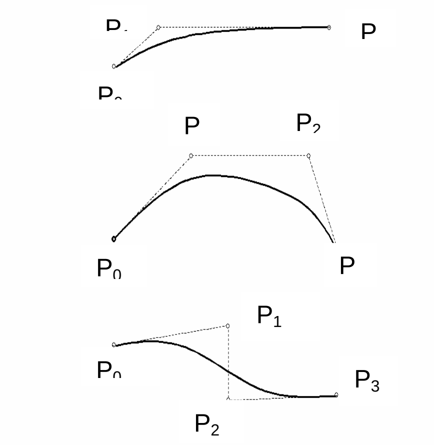


Fig. 4. Bezier curves and control points

$$C(n,i) = \frac{n!}{i!(n-i)!}$$

There are several properties of the Bézier curves that are important for the design of our EC scheme; these are [Farin, 1993]:

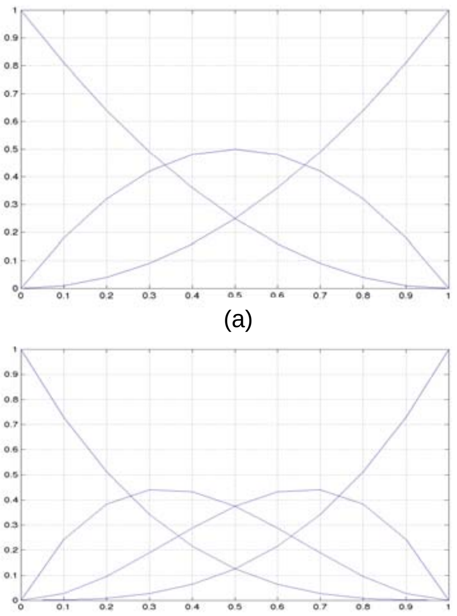
1. Endpoint interpolation: The function interpolates the first and last control points; that is, the curve segment must start on P_0 and end on P_n. This property comes from the fact that the Berstein Polynomials form a partition of unity:

$$\sum_{i=0}^{n} B_i^n(u) \equiv 1$$

This fact can be proved with the help of the binomial theorem:

$$1 = [u + (1-u)]^n \; = \sum_{i=0}^{n} \binom{n}{i} u^i (1-u)^{n-j} \; = \sum_{i=0}^{n} B_i^n(u)$$

Figure 5 shows the blending functions for *n = 2* and *3*.



(a)

(b)

Fig. 5. Berstein polynomials. (a) Three points, n=2; (b) Four points, n=3.

2. Symmetry: It does not matter if the Bézier curves are labeled $P_0, P_1, \ldots, P_n,$ *or* $P_n, P_{n-1}, \ldots, P_0$. The curves that correspond to the two different orderings look the same; they differ only in the direction they are traversed.

3. Linear Precision: This can be expressed as:

$$\sum_{i=0}^{n} \tfrac{i}{n} B_i^n(u) = u$$

When the control points are uniformly distributed on a straight line joining two points *a* and *b*, the curve that is generated is also a straight line.

4. Pseudo-**local control:** The Berstein polynomials B_i^n have only one maximum and attains it at *t=i/n*. If we move only one control point say P_i, then the curve is mostly affected by this change in the region of the curve around the parameter value *i/n*. Roughly speaking, the maximum of each B_i^n is *1/3*; thus the change of P_i by 3 units.

For any Bezier curve, each P_i is weighted by its associated B_i^n; when *u=0*, P_0 is given a weight of 1.0 and P_l through P_n a weight of zero. Less weight is given to P_0 and more to each succeeding P_i as *u* increases; reaching a maximum weight for each P_i then *u* becomes *i/n* (property 4). When *u=1* the effect of all control points is zero, except P_n, which is 1.0. Another important property of the Bézier curve is derived in terms of its derivative at the end points, which can be expressed as:

$$\begin{aligned} &n(P_1 - P_0) && for\ u = 0; \\ &n(P_n - P_{n-1}) && for\ u = 1. \end{aligned}$$

3 Reconstruction Scheme

In our proposed scheme we assume that a lost block (8x8 pixels) is surrounded by all its neighbors (undamaged blocks). In current image/video standards this is difficult to achieve since contiguous blocks are usually inserted in one packet, meaning that the loss of a packet during transmission represents the loss of several blocks (or at least one block). In order to deal with this problem we can make use of block-based interleaving schemes, which transforms the loss of a sequence of block into a random loss. Interested readers are referred to reference to Posnak, et al., 1995, for more information on data interleaving.

Our EC scheme introduces a curvature analysis, average edge direction and higher order interpolation to the scheme presented by Zeng and Liu, 1999. The algorithm can be described in 4 main steps:
1) Extraction of edges, edge directions and curvature around the lost block; 2) Edge matching analysis and determination of the interpolation scheme based on global (edge to edge transition curvature) and local curvature (edge curvature) measurements; 3) Edge modeling based on BC; and 4) Interpolation of low frequency information between the edge structures (Figure 1).

3.1 Edge Detection

One of the main factors that affect the overall performance of directional interpolators is the selection of a good edge detector scheme. All further analysis (curvature extraction and edge matching process) depends on this first step. If the algorithm does not detect a visually active edge, it will be reflected in the final quality of the reconstructed block. On the other hand, if the algorithm detects every single change in luminance, it will complicate the decision process. After careful experimental consideration and for comparison purposes, we decided to use a generalized version of the binary edge detection scheme presented in [Zeng and Liu, 1999]. The generalized binary edge detector works as follows: The *k* largest values LA_k and the *k* smallest values SM_k among the T surrounding layers of the lost block are first determined, for $k=T^2$. Their average value $(L_k + S_k)/2$ is then used as the threshold (producing a bi-level image). A median filter of length 3 is applied after the threshold operation to eliminate isolated white or black points in the binary layers (see Figures 1 and 13). Since edge curvature needs to be computed at each transition point, the number of layers *T* around the lost block has to be $T \geq 3$. Experimentally, *T=4* gave the best results [Hasimoto, 2001].

3.2 Local and Global Curvature

After edge detection, the number of transitions in the inner layer is identified. A transition point indicates that an edge is passing through this point. If more than 2 transition points are detected, outer layers (for *k* > 1) are used for the estimation of both the local and global curvature at each transition point. These estimations are important for the edge coupling (matching) as well as to define the type of directional interpolation needed for the linking process (order of the BC), as shown below. Local edge curvature can be defined by the following equation [Farin, 1993]:

$$k = \frac{\left|\left(d^2y/dx^2\right)\right|}{[1+\left(dy/dx\right)^2]^{3/2}}$$

which represents the curvature of a twice differentiable function *y=f(x)* at a point *(x,y)*. For digital images, this equation is not easy; it requires mapping down the edge starting from the transition point in the inner layer up to the value of *k* (number of layers surrounding the lost blocks). In addition, it may be too sensitive to small variation in pixel direction. Instead, a simpler scheme that analyzes the average behavior of the curvature is proposed, as described next.

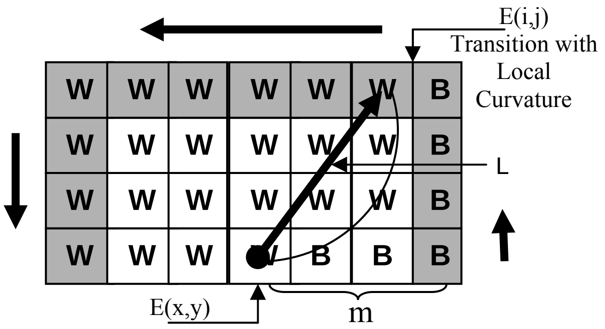


Fig. 6. Computation of local curvature at transition point E(x,y). Shaded pixels represent the searching path (with radius m) for a corresponding transition point. W and B are white and black pixels respectively after the edge detection.

Consider a transition point *E(x,y)* at the upper side of the lost block, as shown in Figure 6. A squared path is traced (shaded pixels), starting at *E(x+m,y)*, and ending at *E(x-m,y)* for *m = T-1*. The difference between the current and next pixels is computed for all points in the path. If there is a point *(i,j)* for $x-m \leq i \leq x+m$ and $y-m \leq j \leq y$, whose value (after the difference) is different than zero, a transition point *E(i,j)* has been found. Otherwise it proceeds with the next pixel in the path. If no transition is found after reaching the end of the square path, *m* is set to *m-1* and the process is repeated as previously described. The transition point *E(i,j)* is connected to *E(x,y)* through a straight line L. The local curvature is approximated by counting the number of white (#W) and black (#B) pixels in the new square region delimited *E(i,j)* and *E(x,y)*, except those falling on L. Curvature values are assigned as follow:

$$k_l = \begin{cases} +1 & if\ \#W > \#B \\ 0 & if\ \#B = \#W \\ -1 & if\ \#B < \#W \end{cases}$$

where k_l is the approximated local curvature.

Global curvature is computed differently than the local curvature. The objective here is to determine the curvature between 2 different edge transitions E_i and E_j connected by a straight line G_{ij} (see Figure 1c). The global curvature is defined as [Zeng and Liu, 1999]:

$$k_g = \phi_i + \phi_j$$

where ϕ_i and ϕ_j are the angles between G_{ij} and the edge at transition points E_i and E_j respectively. The curvature values k_l and k_g are used to define a cost function for the edge matching process (and for the direction of the interpolation as well). The cost function for transition points *i* and *j* is defined as:

$$C_{ij} = w_1 k_g + w_2 \left| k_l^i \phi_i - k_l^j \phi_j \right| \qquad (1)$$

$$\ldots$$

where the w_1 and w_2 are the normalized weights as a function of the local curvature values,

$$w_1 = \frac{\left| \sim k_l^i + \sim k_l^j \right|}{2}, \qquad w_2 = \frac{\left| k_l^i \right| + \left| k_{jl}^i \right|}{2}$$

The symbol ~ represents the logical not (~0 = 1, ~a = 0, for a ≠ 0). If no local curvature is found on the edges under consideration, only the first term is used in the cost function ($C_{ij} = k_g$). Both terms are used with equal weight, if a local curvature is detected in one of the edges; finally, only the second term is used when local curvature is detected in both edges ($C_{ij} = \left| k_l^i \theta_i - k_l^j \theta_j \right|$). This is closely related to the physical meaning of each term in eq. 1. The first term on the right side of eq. 1 represents the contribution of the global curvature to the cost function. It states that changes in edge direction inside a block are smooth (in particular for an 8x8 pixel block, as used in this work); that is if two edges belong together, k_g must takes on small values. When local curvature is significant (high frequency components), the smoothness assumption is not always satisfied, that is the reason of the local curvature term in eq. 1. This term represents the states that local curvature is maintained along the edge; that is if two edges belong together, they will be in general on the same side of the line G_{ij}, with small ϕ_i - ϕ_j. Drastic changes on the edge curvature are penalized by the cost function (Zeng and Liu cost function, depends on k_g only). The complete edge geometry of the missing block is then obtained as follow:

$$\min \sum_{i \neq j} C_{ij} \qquad (2)$$

with the restriction that no edge crossing is allowed..

3.3 Edge Modeling

The next step after the edge detection and matching processes is the edge modeling. The number of possible transitions is divided into 5 cases: 0, 2, 4, 6 and more than 6 transitions on the inner layer. Each transition is analyzed as follow:

1. No-edge transition: If the range of gray levels is less than a specified threshold $T_g = 20$, the missing block is classified as a smooth block. A simple method based on bi-linear transformation can be effectively applied in this kind of situation. In the first method, a missing pixel is reconstructed by the following equation:

$$P = \sum_{i=1}^{4} \hat{d}_i \, g_i$$

where,

$$\hat{d}_i = \frac{d_i}{\sum_{j=1}^{4} d_j}$$

with d_i and g_i as defined in Figure 7.

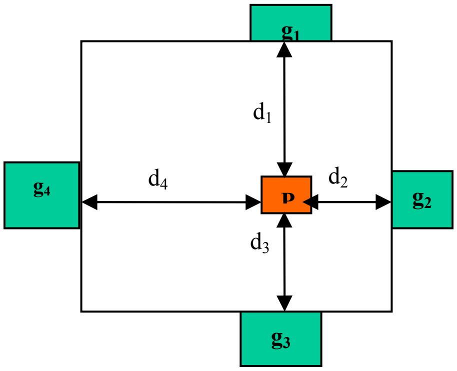


Fig. 7. Bi-linear pixel reconstruction for homogeneous regions (no transition points).

2. Two edge transitions: This is the most common case for the binary edge detector. Since the edges (denoted by E_1 and E_2) have to be linked to each other, the only decision to make is about the interpolation process between them. For this, we use k_g as follow:

- If $\boldsymbol{k_g = 0}$, all pixels are interpolated using bi-linear interpolation along the direction of θ_{12} (angle of G_{12} respect to the x-axis connecting edge 1 and edge 2); that is:

$$P = \sum_{i=1}^{2} \hat{d}_i \, g_i$$

- If $\boldsymbol{k_g > 0}$ **and** $\boldsymbol{\pi - k_g \geq \pi/2}$, all pixels are interpolated using a second or third order Bezier curve depending on the structure of the edge. A fundamental problem here is finding all the points that will minimize the interpolation error.

We found that 3 control points in the BC can represent all possible links between any two edges in an 8x8 pixel region, and rarely 4 are needed. Two of the control points are already known the start (P_0) and end (P_{np-1}) points, where *np* is the number of control points. These are located at the transition points E_1 and E_2 respectively. If we consider a BC with 3 control points, the third control point (P_1) is selected to be the intersection point between the lines L_1 and L_2 (see Figure 8). The point P1 satisfies the derivative properties of the BC defined in section 2.1. Once all the control points are defined, the directional interpolation is performed along the BC, considering only the extreme points as in the case of bi-linear interpolation.

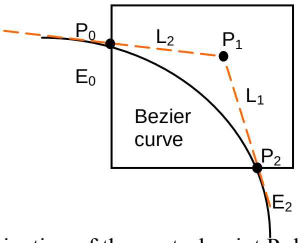


Fig. 8. Determination of the control point P_1 based on the derivatives L_1 and L_2 at points P_0 and P_2 respectively.

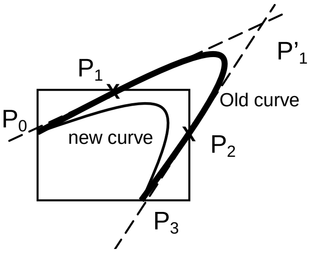


Fig. 9. Avoiding cross-border interpolation in wrong intersection points by setting new control points (cross marks) on the inner layer.

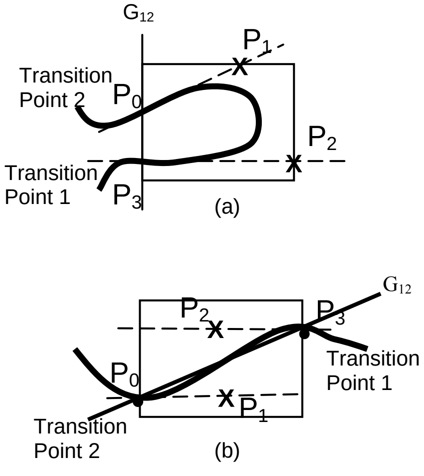


Fig. 10. Bad intersection points. (a) Transition points on the same side respect to G_{12}; and (b) Transition point on the opposite side with respect to G_{12}.

There are some cases that need special attention: a) when the interpolated pixels cross the limits of the lost block (Figure 9), b) when the lines L_1 and L_2 do not intersect or intersect in the same side as L_1 and L_2 (Figure 10a), and c) if L_1 and L_2 are in opposite sides with respect to G_{12} (Figure 10b). The solution in all cases is the inclusion of additional control points in order to increase the curve rate change (curvature). In the first case (a), a new pair of intermediate control points is defined on the intersection of L_1 and L_2 with the inner layer (making a total of 4 control points). We then performed the interpolation of the missing pixels as in the previous cases. For the solution of the second case (b), the distance between each control point and the inner layer is computed along the direction of their respective tangents (L_i). The new control points are positioned at the minimum distance in the direction of their respective tangents (L_1 and L_2). Finally, when the edges are on the opposite side to G (case c), the new control points are $P_1 = [E_{1y}, E_1x+ (E_{2x} – E_1x)/2]$ and $P_2 = [E_{2y}, E_{2x} - (E_{2x} – E_1x)/2]$ for transition point E_1 and E_2 respectively. If the new control points are off the block limits in either or both control points they are set to the inner layer.

- If **k_g > 0 and л-k_g < л/3** the decision is up to the local curvature at each edge. If $k_l^i = 0$ and $k_l^j = 0$, it is assumed that a corner exist, and bilinear interpolation is performed using 2 or 4 extreme points along L_1 and L_2, as shown in Figure 11. Otherwise ($k_l^i \neq 0$ or $k_l^j \neq 0$) BC is used for the interpolation process.

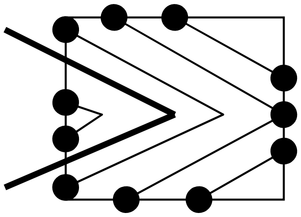


Fig. 11. Linear interpolation using 2 extreme points (•) for the reconstruction of the missing pixels.

3. Four edge transitions: A decision process for matching edges based on eq.1 is needed here. Let us denote the transition points as 1, 2, 3, and 4, in a clockwise direction along the inner layer. The direction of the interpolation is obtained by calculating the $\min(C_{12}, C_{14}, C_{23}, C_{34})$. Once the first edge couple is defined, the second is automatically known. Because *no edge crossing is allowed*, C_{13} and C_{24} are not possible combinations. Next, the interpolation algorithm (BC or bi-linear interpolation) is decided as in the case of the two-edge transition points. Four transition points divides the original missing block into three regions, as depicted in Figure 12. Missing pixels in region 1 and 3 are interpolated using BC and bi-linear interpolation, which represent the interpolation schemes for E_{14} and E_{23} respectively. Region 2 is influenced by both BC and bi-linear interpolation schemes, therefore the missing pixels closer to region 1 are interpolated using BC, while those pixels closer to region 3 are interpolated using bi-linear interpolation.

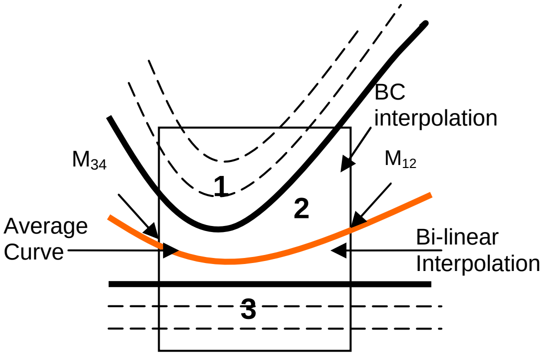


Fig. 12. Two-edge transition interpolation scheme. Missing pixels in region 2 are interpolated using interpolation schemes in region 1 and 3. The average curve marks the limit between both schemes.

The transition curve between the two interpolation schemes in region 2 is found as follow:

- A middle point (M_{ij}) is calculated between the unmatched transition points (see Figure 12).
- Let $L_{34} = (L_3+L_4)/2$ and $L_{12} = (L_1+L_2)/2$ represent the average tangent between the unmatched edges at the middle points M_{43} and M_{12} respectively. A second order BC is obtained using the extreme points M_{43} and M_{12} and the third control point P, which is obtained as depicted in Figure 8. Missing pixels along this curve are then interpolated.

A simpler approximation not requiring the computation of a new BC can be used instead. Region 2 can be interpolated until BC in region 1 and bi-linear interpolation in region 3 meet each other. The remaining missing pixels can be interpolated linearly using the surrounding information.

4. Six edge transitions: A pair of transition points with the smallest C_{ij} is identified. The rest of the procedure is similar to step 3, except that all missing pixels within a region are interpolated using bi-linear interpolation instead of a combined interpolation process. There are some important remarks that need to be pointed out for this case. Because of the assumption that no edge crossing is allowed, we have 9 possible pairs of transition described by the odd number |i-j| odd. For some selected pairs of transition, the other four are fixed. For example, if the transition points 1 and 4 have been selected, then the transition point 2 has to be connected to point 3, and transition point 5 has to be connected to transition point 6.

5. Six+ edge transitions: For an 8x8 pixel block, more than six transitions are seen very infrequently. In this case, we concentrate on finding the most dominants direction using the gradient operator over the two innermost layers. Regions with similar edge direction are bi-linearly interpolated.

When the number of edge transition points is odd, the interpolation of the paired edge transitions is performed first. The stand alone transition is just elongated to the center of the missing block at the most, without getting across any other pair of edge transition interpolation.

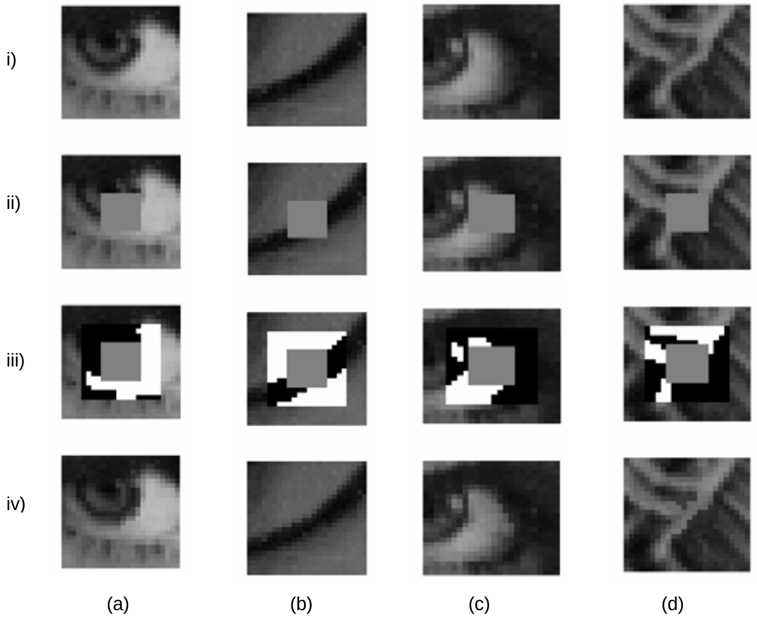


Fig. 13. Block reconstruction (8x8 pixels) using a second degree Bezier polynomial for a) 2, b) and c) 4, and d) 6 edge transition points. Row sub-images (16x16 pixels) represent: i) original data, ii) damaged block, iii) detected transition points, and iv) interpolated block.

4 Results

The performance of the described scheme is shown in Figure 13 for 2, 4, and 6 transition points around the lost block (8x8 pixels) using a second degree BC (3 control points). Results are presented in a 16x16 pixel sub-image of Lenna with the following format: i) original, ii) damaged, iii) detected binary edge, and iv) reconstructed sub-image. Figure 13a shows a missing block with visually 3 edge regions, one in the pupil, another in the iris and other in the sclera (white area of the eye). Because we are using a binary edge detector (see section 2), only two regions were detected (sclera and the rest). Despite of this, our scheme was able to reproduce the natural curvature of the eye (different from Zeng and Liu's method, see Figure 3 for comparison), as well as to almost perfectly recover all the three regions in the interpolated area. The reason for this is that we use the same BC to directionally interpolate all the information of the missing block (including the low frequency components). In situations with more than 2 transition points around the lost block, the use of the cost function defined in eq.2 is required for the edge matching process. If the edge matching process fails to reconstruct the original structure of the lost block, it is very likely that the reconstruction process will give the wrong result. With this regard, the proposed (improved) cost function showed an outstanding performance as shown in Figures 13(b-c) and 13d corresponding to 4 and 6 transition points. At normal resolution (entire Lenna image) the reconstructed and original image are indistinguishable, at small detail the differences are due to small variations in the tangents of the linked edges.

5 Conclusions

We have proposed a new geometric error concealment scheme, which takes the edge direction and edge curvature (local and global) as boundary conditions to generate an n-degree BC (for $n \leq 3$). The generated BC is used for the reconstruction of the missing block information. The proposed scheme has shown its ability in reconstructing complex high frequency information and high degree of accuracy during the edge transition matching (coupling) process.

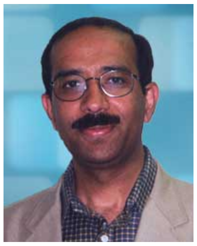
Acknowledgement
This work was supported by CONCYTEG (Consejo Estatal de Ciencia y Tecnología de Guanjuato) under grant number 04-02-K117-020.

References

1. **Farin G.,** "Curves and Surfaces for Computer Aided Geometric Design". *Computer Science and Scientific Computation*, Academic Press, Inc., New York, NY, 1993.
2. **Hasimoto, R. B.**, "Error Resilient Framework for Image/Video Transmission over Packet Switch Networks". Ph.D Thesis, *University of Delaware*, Sept. 2001.
3. **Hemami, S. S. and T. H.-Y Meng,** "Transform Coded Image Reconstruction Exploiting Interblock Correlation". *IEEE Trans. Image Processing*, Vol. 4, pp-1023-1027, July 1995.
4. **Jung K.-H., J.-H. Chang, and W. Lee,** "Error Concealment Using Projection for Block-Based Image Coding". *SPIE* Vol. 2308, pp. 1466-1476, 1994.
5. **Kwok W. and H. Sun,** "Multidirectional Interpolation for Spatial Error Concealment. IEEE Trans. Consumer Electron., Vol. 39, pp. 455-460., August 1993.
6. **Pei-Chun C. T., and T. Chen,** "Second-Generation Error Concealment for Video Transport Over Error Prone Channels. *ICIP* 2002.
7. **Posnak E. J., S. P. Gallindo, A.P. Stephens, and H.M. Vin,** Techniques for Resilient Transmission of JPEG Video Streams. *Proc. of Multimedia Computing and Networking*, pp. 243-252, February 1995.
8. **Salama P., N. B. Shroff, and E. J. Delp,** "Error Concealment in Encoded Video", *IEEE Journal on Selected Areas in Communications.* , Vol. 18, No. 6, June 2000, pp. 1129 -114.
9. **Shahram, S., Kossentini F., and Ward R.**, An Adaptive Markov Random Field Based Error Concealment Method for Video Communication in an Error Prone Environment. *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)* 1999.

10. **Wang Y., Q-F. Zhu, and L. Shaw,** "Maximally Smooth Recovery in Transform Coding. *IEEE Trans. Commun.*, Vol. 41, No. 10, 1993, pp. 1544-1551.
11. **Zeng W. and B. Liu,** "Geometric-Structure-based Error Concealment with Novel Applications in Block-based Low Bit Rate Coding". *IEEE Trans. Circuits and Systems for Video Tech.*, June 1999.
12. **Zhu Q.-F., Y. Wang, and L. Shaw,** Coding and Cell Loss Recovery for DCT-based Packet Video. *IEEE Trans. Circuits Syst. Video Tech.*, Vol. 3, pp. 248-258, June 1993.

Rogelio Hasimoto Beltrán received his Ph.D. degree from the University of Delaware, USA, in 2001. After his Ph.D, he spent two years at Akamai Tech. as a Senior Software Engineer. Since 2003 he has served as an associate researcher in the Department of Computer Science at the Center for Research in Mathematics (CIMAT), México. His research interests include image/video processing and compression, robust multimedia transmission and multimedia encryption. He is member of the IEEE association.



Ashfaq A. Khokhar received his Ph.D. in computer engineering from University of Southern California, in 1993. He is in the Department of Computer Science and Department of Electrical and Computer Engineering of the UIC, where he currently serves on the rank of Professor. Dr. Khokhar has published over 100 technical papers and book chapters in refereed conferences and journals in the area of wireless networks, multimedia systems, data mining, and high performance computing.