SPATIAL ERROR CONCEALMENT BASED ON BEZIER CURVES

*Rogelio Hasimoto-Beltrán and **Ashfaq A. Khokhar

*Department of Computer Sciences,
Center for Research in Mathematics
Guanajuato, Gto. 36000,
Mexico
hasimoto@cimat.mx

**Departments of CS/ECE
University of Illinois
Chicago, IL 19013,
USA
ashfaq@uic.edu

ABSTRACT

This paper presents a geometric error concealment scheme for DCT-based image/video data based on the Bezier polynomials (BP). The proposed scheme 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 BP. The resulting BP is then used to interpolate the edges of the lost block, as well as to directionally reconstruct its low frequency data. When multiple edges are identified crossing the lost area, a cost function defined in terms of the local and global edge curvatures is used to find the best match for each missing edge. Experimental results show that our approach provides near perfect reconstruction and excellent subjective quality of the restored data, outperforming current linear interpolation schemes in the literature.

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 ([2]). 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 ([3]). 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 two main approaches have been proposed in the literature for recovering from transmission errors ([3], [4]): Minimum Distance Interpolators (MDI) and Directional Interpolators (DI). The first category makes use of the smoothness property among neighboring blocks, and the best reconstruction is achieved by an energy constrained minimization approach ([5]-[7]). 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. EC schemes by directional interpolation are based on the extraction of the local structure (e.g. direction of edges) around the missing block for its reconstruction [8], [9]. These techniques have become very popular because of their simplicity and efficiency to recover low and high frequency components. Kwok and Sun [8] proposed a spatially correlated edge information scheme to perform a multidirectional interpolation (along 8 different paths) to restore the missing blocks. Jung, et.al, [9], developed an algorithm similar to [8], 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, [1], 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.

In general, the above techniques give relatively good results under simple edge patterns, such as straight line crossing throughout the lost block. Under more complicated structures they fail to reconstruct the original detail (see Figure 1). In this work, we propose a generalized interpolation scheme that is capable of representing more complicated edge geometries (such as curved edges, rounded corners, etc.) by considering the average edge direction, local as well as global edge curvatures. We use adaptive Bèzier polynomials to model the geometric structure of the lost blocks.

This paper is organized as follows. Section 2 describes the theory behind our proposed model, and presents a simplified scheme for the computation of the local edge curvature. Section 3 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 4.

2. BÉZIER CURVES BASED ERROR CONCELAMENT

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 2 [10]. Once the control points are given, the curve shape is determined. This behavior of the BC is an important characteristic useful to realize efficient error concealment schemes. When a block is lost, the

This work was supported by SEP/SEBYN-CONACYT under grant reference 107.

0-7803-9332-5/05/$20.00 ©2005 IEEE

average features such as edge position and average tangent of the surrounding blocks can be used in the BC model as control points (edge curvature is used for the edge matching process). The model is then used to determine the best fitting curve under the computed boundary conditions. Another characteristic of the BC is that it does not require additional dependencies of the control points in the lost block, as in the case of the spline curve model, the dependencies are fixed by the model. Therefore, the key issue is to define the best properties representing the lost block, in order to minimize the difference between the restored block and the neighboring blocks. This is accomplished by setting additional constrains as explained later on this section.

The Bézier 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 2), 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

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

Proposed Reconstruction Scheme

Our EC scheme improves upon the technique proposed in [1] by introducing curvature analysis, average edge direction and higher order interpolation in the reconstruction process. The proposed reconstruction scheme consists of the following four 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.
4) Interpolation of low frequency information between the edge structures.

One of the main factors that affect the overall performance of directional interpolators is the selection of a good edge detector. The performance of the subsequent steps depends on this selection. 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 evaluations, we have elected to use a generalized version of the binary edge detection scheme presented in [1].

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. A median filter is applied after the threshold operation to eliminate isolated white or black points in the binary layers. 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 [3].

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 curvature can be defined by the following equation [10]:

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

It 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.

Consider a transition point $E(x,y)$ at the upper side of the lost block, as shown in Figure 3. 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}. The global curvature is defined as:

$$k_g = \phi_i + \phi_j$$

where ϕ_i and ϕ_j are the angles of 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:

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

where the w_1 and w_2 are the normalized weights in 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 contribution of the local curvature to the edge matching process. It 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 $\phi_i - \phi_j$. Extreme changes on the edge curvature are penalized by the cost function (Zhen and Liu cost function [1], 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..

The next step after the edge detection and matching processes is the edge modeling. Edge pixels are interpolated using a second or third degree Bézier 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 points are needed. Two of the control points, i.e., the start (P_0) and end (P_{np-1}) points, are already known where *np* is the number of control points. These are located at the transition points E_0 and E_1 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 2). Once all the control points are defined, the directional interpolation is performed along the curve, considering only the extreme points as in the case of bi-linear interpolation.

3. SIMULATION RESULTS

The performance of the proposed scheme is shown in Figure 4 for 2, 4, and 6 transition points around the lost block (8x8 pixels). using a second degree BP (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 4a 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). Since 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 1 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 BP 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 3b and c 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.

4. CONCLUSIONS

We have proposed an efficient geometric error concealment scheme based on a *n*-degree Bézier Polynomial (for $n \leq 3$). The proposed scheme 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 BP. Based on experimental results on sample eye images, the scheme has shown its superior performance in reconstructing complex high frequency information, and achieves high degree of accuracy during the edge transition matching (coupling) process.

5. REFERENCES

[1] W. Zeng 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.

[2] P. Salama, N. B. Shroff, and E. J. Delp, "Error Concealment in Encoded Video". www.ece.purdue.edu/~ace.

[3] R. Hasimoto-Beltrán, "Error Resilient Framework for Image/Video Transmission over Packet Switch Networks". Ph.D Thesis, *University of Delaware*, Sept. 2000.

[4] C. T. Pei-Chun, and T. Chen, "Second-Generation Error Concealment for Video Transport Over Error Prone Channels. *ICIP* 2002.

[5] Q.-F. Zhu, 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.

[6] Y. Wang, Q-F. Zhu, and L. Shaw, "Maximally Smooth Recovery in Transform Coding. *IEEE Trans. Commun*., Vol. 41, pp. 1544-1551.

[7] S. S. Hemami and T. H.-Y Meng, "Transform Coded Image Reconstruction Exploiting Interblock Correlation". *IEEE Trans. Image Processing*, Vol. 4, pp-1023-1027, July 1995.

[8] W. Kwok and H. Sun, "Multidirectional Interpolation for Spatial Error Concealment. IEEE Trans. Consumer Electron., Vol. 39, pp. 455-460., August 1993.

[9] K.-H. Jung, J.-H. Chang, and W. Lee, "Error Concealment Using Projection for Block-Based Image Coding". *SPIE* Vol. 2308, pp. 1466-1476, 1994.

[10] G. Farin, "Curves and Surfaces for Computer Aided Geometric Design". *Computer Science and Scientific Computation*, Academic Press, Inc., New York, NY, 1993.

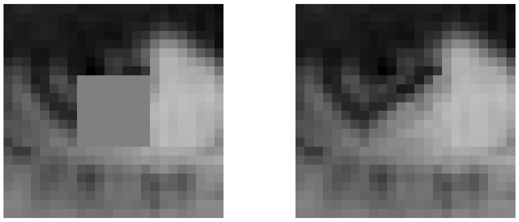


Figure 1: Lost block (left) and its reconstruction based on [1] (right).

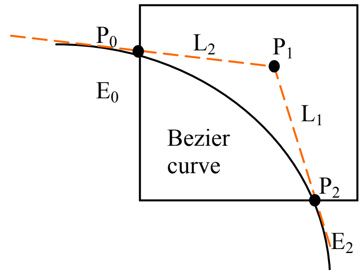


Figure 2. Determination of control point P_1 based on the derivatives L_1 and L_2 at points P_0 and P_2 respectively. E_0 and E_2 represent the transition points around the lost block.

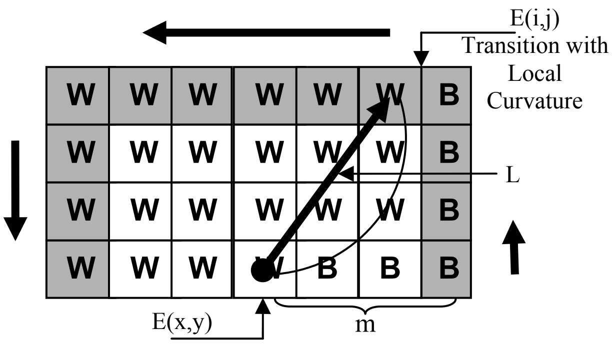
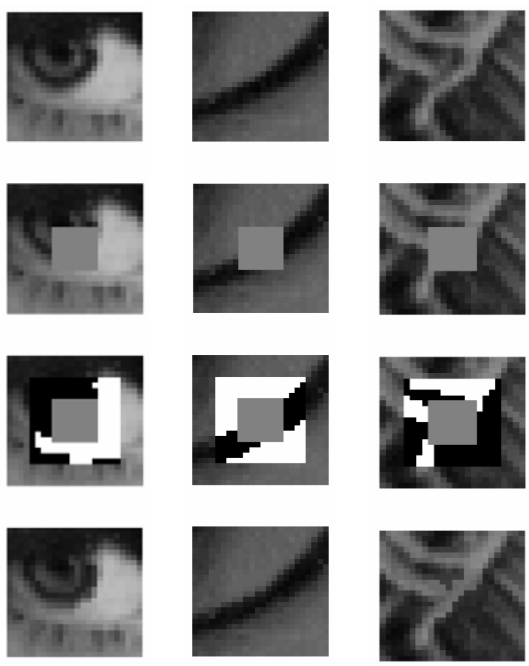


Figure 3. Computation of local curvature at transition point E(x,y). Shaded pixels represent the searching path (with radius m) for a corresponding transition point. W and B are white and black pixels respectively after the edge detection.

i)
ii)
iii)
iv)



(a) (b) (c)

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