# ERROR RECOVERY OF IMAGE-BASED DEPTH MAPS USING BÉZIER CURVE FITTING

*S. Marcelino*[1,2], *P. Assuncao*[1,4], *Sérgio M. M. de Faria*[1,4], *S. Soares*[2,3],

[1]Instituto de Telecomunicações; [2] Universidade de Trás-os-Montes e Alto Douro; [3]IEETA;
[4]Instituto Politécnico Leiria, ESTG Portugal

*stam@co.it.pt, {pedro.assuncao, sergio.faria}@ipleiria.pt, salblues@utad.pt*

**ABSTRACT**

This paper proposes a method to recover lost regions in image-based depth maps used in video plus depth 3D format. This method performs depth maps reconstruction taking into account depth contours within the lost regions. This is achieved by extracting the contours and recovering their lost segments based on Bézier curve fitting, followed by spatial interpolation. The proposed method maintains contour smoothness and uses them as the boundary limits of homogeneous depth regions, which are then filled through weighted pixel interpolation. The experimental results show that the proposed method yields better synthesized images than classic spatial concealment methods, uniquely based on pixel interpolation techniques. The method presented in this paper is able to outperform the reference method, in terms of PSNR by up to 1.91dB. The subjective quality is also shown as being significantly better.

***Index Terms—*** Error concealment, depth map, Bézier curves.

## 1. INTRODUCTION

Nowadays 3D content is spreading fast and is now available to the general public, significantly increasing the number of services and applications, such as 3D movies and 3DTV which have reached great popularity. However, there are still several problems and research challenges to be addressed before an acceptable quality is guaranteed in all delivery services and applications.

The most common 3D video coding formats are the MVC (Multiview Video coding) [1], and the MVD (Multi-view video plus depth) [2]. In the MVC format, all available views are explicitly encoded, each one exploiting the redundancy in temporal domain and among views. In MVD, each view has an associated depth map, allowing other views to be synthesized using depth-based image rendering (DIBR) [3] The main advantages of the latter format are the possibility of generating multiple views from only one view, and the significant bitrate savings that can be achieved due to the use of a depth map. The depth map is represented as a grayscale picture with the same number of pixels as the corresponding color image.

In the case of the video+depth representation, the quality of depth perception experienced by users might be seriously affected by the integrity of the depth map. For example, the authors in [4] show that quality enhancement of a depth map can improve the rendering quality significantly. Thus, in the case of transmission errors, proper error concealment of the depth map is of major importance in the 3D quality delivered to the users.

The problem of depth map error recovery addressed in this paper is currently an active field of research, but there still are several limitations to overcome. For instance, the method proposed in [5] assumes that in presence of depth map errors, the corresponding color image region is always received with no errors, in order to be used in the depth map recovery. Although the authors report good objective results, the impact on the synthesis quality of other views is not evaluated.

The authors in [6] implemented a method to conceal lost frames in the depth maps by exploiting the correlation of motion activity between the color image and the depth map. Good objective results are also reported, but some block artifacts are subjectively noticeable in the depth map, mainly at the edges between the foreground and the background. Thus, most of the error recovery techniques published so far for coping with missing data in depth maps are mainly based on information from temporally adjacent depth maps and/or corresponding color image. This paper proposes a rather different method based on the combination of depth contour recovery with depth spatial information located in the surrounding areas of the corrupted region. This has the advantage of being useful for both image and video depth map recovery, when no temporal information is available.

In the proposed method, depth map contours representing sharp transitions between different levels of depth, are reconstructed by using curve fitting techniques based on Bézier curves [7]. Firstly, all contours representing sharp transitions in depth values are extracted from the received depth map. Secondly, the depth lost blocks are classified in two categories: non-edge lost blocks and edge lost blocks. For the non-edge lost blocks, weighted pixel interpolation is used to compute the values of the missing depth pixels, while for edge lost blocks the missing edge/contour is first reconstructed by using Bézier curves. Based on the recovered depth contours, the depth pixel values inside these edge lost blocks are also computed using weighted pixel interpolation. In this case the contours are used as boundaries to distinguish between regions with different levels of depth. Bézier curves were previously used in image/video concealment methods [8], but their application in error recovery of depth maps cannot be found in the literature.

This paper is organized as follows. Section 2 presents the proposed method for error recovery in image-based depth maps. Experimental results are shown and discussed in Section 3, while Section 4 concludes the paper.

## 2. PROPOSED METHOD DESCRIPTION

The proposed algorithm consists in four main steps, as shown in Fig. 1. Firstly, all contours are extracted from the depth map. Secondly, the contour around each lost area is analysed to find which end-points should be connected together. Thirdly, based on the end-points pairs, matched together on the previous step, an additional pair of control-points is computed, and then reconstruction of the contour with a Bézier curve is carried out. Finally, all lost blocks are

This work has been supported by the Fundação para a Ciência e a Tecnologia, Portugal (under the PhD Grant SFRH/BD/64988/2009 and Project PTDC/EEA-TEL 114487/2009)

concealed using weighted pixel interpolation.

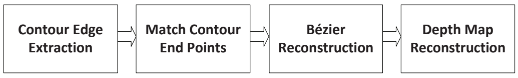



**Fig. 1**: Depth map concealment algorithm

### 2.1. Contour extraction

Contour extraction is based on the variance of a sliding window with size of 3x3 pixels. Firstly, the variance of such window is computed for the whole image. Secondly, the contours are defined as those pixels for which the corresponding variance is above a pre-defined threshold. A value of 5 was empirically found as good tradeoff between sharpness of depth transitions across edges and contour relevance in the lost region. Figure 2, shows an example of a depth map and its respective extracted contour.

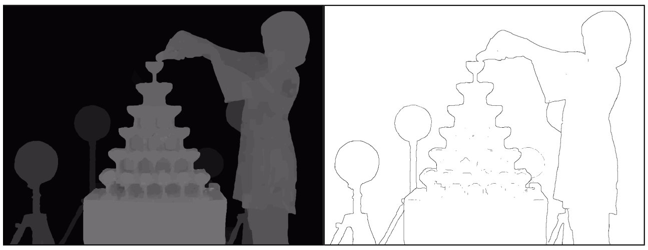


**Fig. 2**: Depth map and respective contour

### 2.2. Matching end-points

Since several different contour segments might be within the lost region, in order to reconstruct them separately, the corresponding end-points should be found and then matched together in pairs for reconstructing the lost contour segments. Such pair of end-points correspond to the contour points (i.e. depth pixels), where the contour is broken by the missing region.

Figure 3, shows an example of a contour passing through a lost area, where $P1$ and $P4$ are the end-points of Contour 1 and Contour 2, respectively. $\beta_1$ and $\beta_2$ are the angles between the tangent vectors (at the end-points) and the straight line $\overline{P1 - P4}$ connecting the two end-points. Reliable and precise computation of $\beta_1$ and $\beta_2$ requires accurate tangent vectors at the end-points. In this work up to 5 contour depth pixels, located behind each end-point, are used to define the tangent vectors.

Matching the end-points is done by comparing $\beta_1$ and $\beta_2$ for all possible combinations of end-points pairs in each lost region, and then by choosing the minimum sum of the corresponding angles $\beta_1$ and $\beta_2$. If different matching pairs give rise to intersecting contours, then such matching pairs will not be considered valid, and another possible matching will be checked.

### 2.3. Contour reconstruction with Bézier curves

The cubic Bézier curves used to connect the lost contour segments are based on four control-points $P1...P4$, where the first ($P1$) and the last ($P4$) of these points are the end-points of the lost contour segment. The control points $P2$ and $P3$ are computed based on the tangent vectors at the end-points $P1$ and $P4$. The first step to find the control points $P2$ and $P3$ is to draw two virtual lines passing at P1 and P4, with angles $\beta_1$ and $\beta_2$, respectively. Then $P2$ and $P3$ are the points located at the middle of segments $\overline{P1 - A}$ and $\overline{P4 - A}$, where A is the intersection point of the two virtual lines (see figure 3). Finally, these four control points are used to define the Bézier curve, which in turn is used to reconstruct the lost segment of the contour.

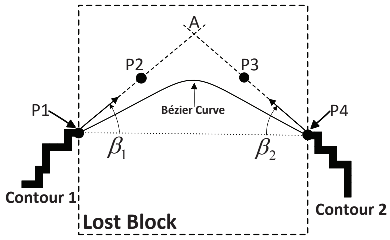



**Fig. 3**: Match lost contours and Bézier control points

A parametric Bézier curve is represented by $Q(t)$, where $t \in [0, 1]$ and $P_1...P_4$ are the 4 control-points described above.

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t)P_3 + t^3 P_4 \quad (1)$$

The Bézier curves $x(t)$ and $y(t)$ representing $Q(t)$ are:

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x && (2) \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y && (3) \\ & && (4) \end{aligned}$$

Defining $C$ as the coefficient matrix of $x(t)$ and $y(t)$, and $F$ as the parameter vector of the parametric curves, i.e.,

$$\begin{aligned} C &= \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \end{bmatrix} && (5) \\ F &= \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}^T && (6) \end{aligned}$$

$Q(t)$ can also be written in matrix form as follows:

$$Q(t) = [x(t) \;\; y(t)]^T = C.F \quad (7)$$

Since equation 7 does not depend on any control point, in order to find the Bézier curve that represents the lost segment of the depth contour, one has to define an equation where P1...P4 are included. This can be obtained by using the definition of the Bézier curve based on the basis matrix $B$ (definition of $B$ can be found in [7]), i.e.,

$$Q(t) = P_i.B.F \quad (8)$$

where the four control-points $P_i$ with $i = 1...4$, are defined by the corresponding coordinates $x_i$ and $y_i$, i.e.,

$$P_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}, i = 1, 2, 3, 4 \quad (9)$$

After substituting $Q(t)$ in equation 8 by $C.F$ from equation 7, the problem solution consists of finding the elements of matrix $C$. The result is the following set of equations:

$$ax = -P1_x + 3 \times P2_x - 3 \times P3_x + P4_x \quad (10)$$
$$ay = -P1_y + 3 \times P2_y - 3 \times P3_y + P4_y \quad (11)$$
$$bx = 3 \times P1_x - 6 \times P2_x + 3 \times P3_x \quad (12)$$
$$by = 3 \times P1_y - 6 \times P2_y + 3 \times P3_y \quad (13)$$
$$cx = -3 \times P1_x + 3 \times P2_x \quad (14)$$
$$cy = -3 \times P1_y + 3 \times P2_y \quad (15)$$
$$dx = P1_x \quad (16)$$
$$dy = P1_y \quad (17)$$

After computing matrix C, the parametric Bézier curves $x(t)$ and $y(t)$ are completely defined. Then, the interval of t ([0,1]) is divided by twice the number of depth pixels comprising the segment $\overline{P1 - P4}$, in order to obtain the number of points of the Bézier curve. Finally, based on the computed curve, the closest depth pixels inside the lost region are chosen to build to the reconstructed contour.

### 2.4. Depth map reconstruction

After recovering all lost contour segments, the depth pixels within a lost region of the depth map must be filled. As mentioned before, every lost block is classified according to whether a lost contour segment is included. The first category includes those blocks that have contours passing through them, while the other category includes the remaining blocks.

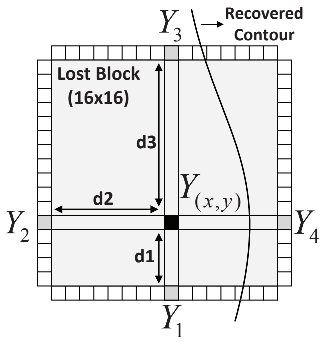



**Fig. 4**: Weighted interpolation

Figure 4 shows the process of finding the value of a depth pixel $Y_{(x,y)}$ inside the lost block, by using weighted pixel interpolation based on four depth pixels ($Y_i, i = 1...N$ and $N = 4$) located in the boundaries of the lost block. Depending on the category of the lost block, not all of the adjacent depth pixels can be used. If a lost contour segment is reconstructed across a block, then there exist adjacent depth pixels located on the opposite side of the contour that cannot be used, because they belong to a very different depth level. In the example of figure 4, pixel $Y_4$ is not used for interpolating $Y_{(x,y)}$ because of the reasons explained above.

The weighted pixel interpolation is implemented by computing a weighted average of the adjacent pixels, as defined by equation 18:

$$Y(x,y) = \frac{\sum_{i=1}^{N} Y_i \times [15 - d_i]}{\sum_{i=1}^{N} d_i}, \quad (18)$$

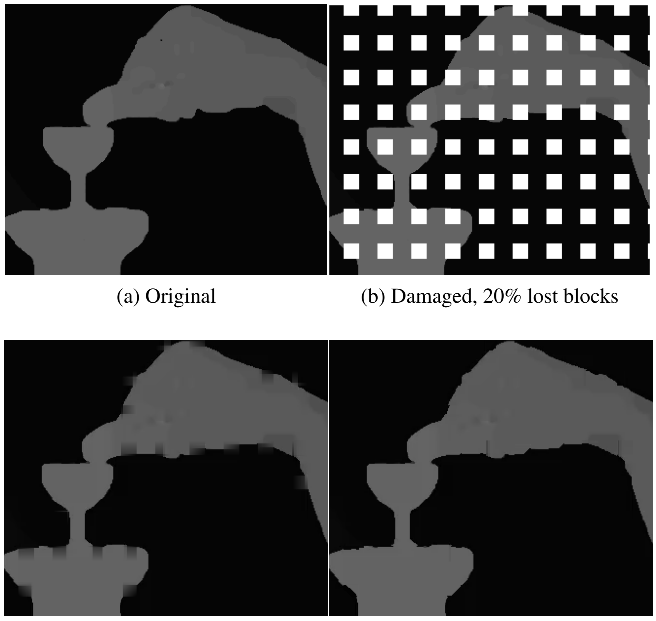
where $d_i$ is the pixel distance between $Y_{(x,y)}$ and $Y_i$, which can have a maximum value of 15 for a 16x16 block size.

## 3. EXPERIMENTAL RESULTS

The performance was evaluated by using the first frame of the five sequences presented in table 1. Note that, since the proposed method is a spatial error concealment algorithm, any other single frame could be used in the experiments. The color images and their corresponding depth maps were encoded with H.264/AVC intra mode and fixed QP=28. The reference software JM17 was used [9]. The PSNR obtained for color and depth is shown in table 1. These color and depth maps were used to evaluate the quality of the second view synthesis, after recovering from depth map errors.

| Sequences | | Resolution | Color (PSNR) | Depth (PSNR) |
|---|---|---|---|---|
| S1 | Ballet | 1024×768 | 41.04 | 46.25 |
| S2 | Breakdancers | 1024×768 | 40.05 | 48.08 |
| S3 | Book Arrival | 1024×768 | 40.50 | 45.89 |
| S4 | Champ. Tower | 1280×960 | 43.28 | 49.6 |
| S5 | Beergarden | 1920×1080 | 38.76 | 48.22 |

**Table 1**: Tested sequences (dB)



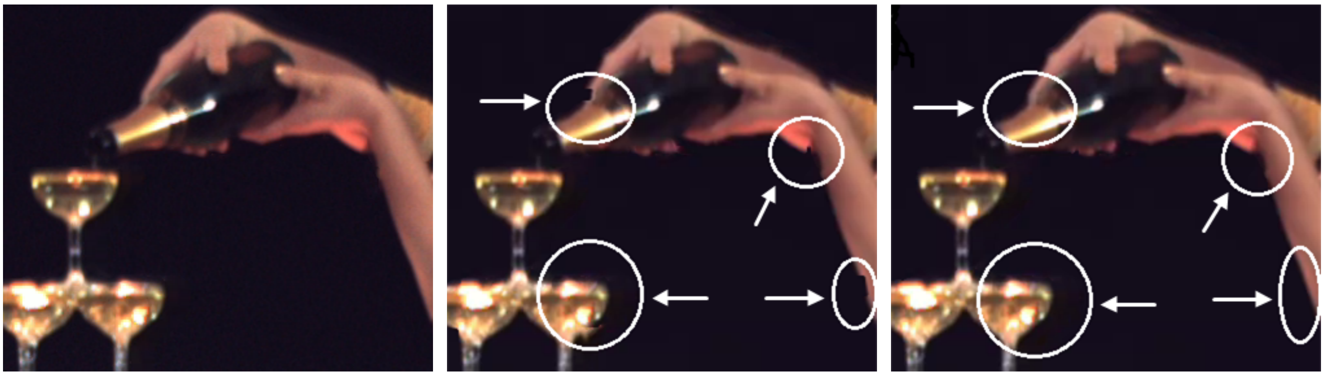
(a) Original (b) Damaged, 20% lost blocks

(c) Recovered with ref. method (d) Recovered with prop. method

**Fig. 5**: Depth Maps

The VSRS (View Synthesis Reference Software) of the MPEG group, was used to synthesize the second view from the base view plus the recovered depth maps [10]. A regular error pattern was introduced in the whole image, in order to obtain a large number of different types of lost contour segments. Thus, depth map losses are not constrained within any particular type of image region. The error pattern was defined as 16×16 lost blocks equally spaced between them, as shown in figure 5b). This type of error pattern may occur from transmission error(namely burst errors) in video streams coded with Flexible Macroblock Ordering (FMO) [11].

The simulation tests where done for three different block loss rates: 5%, 10% and 20%. Performance evaluation was carried out by measuring the PSNR of the synthesised views using the concealed depth maps. A spatial concealment method only based on weighted pixel interpolation, without taking into account the depth map contours, is used as reference for comparison (Ref.) with the proposed



(a) Original

(b) Reference method

(c) Proposed method

**Fig. 6**: Synthesized View

one (Prop.). Since the inpainting process of VSRS for occlusions has a major impact on the PSNR [12], the occluded regions were not included in the PSNR computation of the synthesised view. Therefore, the PSNR obtained for these views is exclusively due to the performance of the depth error recovery methods and does not suffer the influence of the inpainting process.

| Seq. | No errors | 5% Loss | | 10% Loss | | 20% Loss | |
|---|---|---|---|---|---|---|---|
| | | Ref. | Prop. | Ref. | Prop. | Ref. | Prop. |
| S1 | 36.07 | 34.85 | 35.55 | 34.77 | 35.89 | 33.93 | 35.64 |
| S2 | 38.93 | 38.86 | 38.87 | 38.81 | 38.87 | 38.28 | 38.69 |
| S3 | 40.07 | 39.73 | 39.88 | 39.51 | 39.74 | 38.11 | 38.64 |
| S4 | 39.58 | 37.97 | 39.29 | 36.67 | 39.17 | 36.11 | 38.02 |
| S5 | 34.60 | 34.15 | 34.39 | 33.83 | 34.22 | 33.58 | 34.03 |

**Table 2**: Synthesis results (dB)

Table 2 shows the PSNR obtained from synthesised views for the test sequences, at 5%, 10% and 20% block loss rates. As shown in the table, the proposed method outperforms the reference method for all sequence images. A relevant observation in these results is the fact that PSNR gains of the proposed method are generally higher for higher loss rates, which proves its effectiveness.

These results can be subjectively confirmed by observing the details shown in figures 6a to 6c. Observing the details of the synthesized view, obtained with the proposed method (Fig. 6c), one can clearly perceive that indicated regions are much better reconstructed than in Fig. 6b. This is because the proposed method first reconstructs the lost edges of the depth map and then pixel interpolation takes into account the boundaries defined by these edges. Note that these regions correspond to sharp depth transitions with significant impact on the perceived 3D quality. Moreover, by using the proposed method, the block effect that is visible in reference method is almost eliminated.

## 4. CONCLUSION

The experimental results show that good quality depth maps are of major importance to have good quality synthesised images. This paper shows that good quality results can be achieved in synthesised views using error recovered depth maps. By reconstructing the most significant edges in the depth map using Bézier curves, the proposed method is able to yield better 3D quality than the reference one. This method has potential applications in 3D image and video communications over error-prone channels.

## 5. REFERENCES

[1] Y. Chen, Y.-K. Wang, K. Ugur, M. Hannuksela, J. Lainema and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. ASP.*, vol. 2009, pp. 8:1–8:13, January 2008.

[2] A. Smolic, K. Mueller, P. Merkle, P. Kauff, and T. Wiegand, "An overview of available and emerging 3D video formats and depth enhanced stereo as efficient generic solution," in *PCS 2009*, May 2009, pp. 1 –4.

[3] K. Müller, A. Smolic, Kristina Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3D video systems," 2008.

[4] E. Ekmekcioglu, V. Velisavljevic, and S. T. Worrall, "Content adaptive enhancement of multi-view depth maps for free viewpoint video," *Selected Topics in Signal Processing, IEEE Journal of*, 2010.

[5] Y. Liu, J. Wang, and H. Zhang, "Depth image-based temporal error concealment for 3-D video transmission," *CSVT, IEEE Transactions on*, vol. 20, no. 4, pp. 600 –604, 2010.

[6] C.T.E.R. Hewage, S.T. Worrall, S. Dogan, and A.M. Kondoz, "A novel frame concealment method for depth maps using corresponding colour motion vectors," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2008*, May 2008, pp. 149 –152.

[7] Feiner Hughes Foley, Andries van Dam, *Computer Gaphics, Principles and Practice*, Addison Wesley, 1990.

[8] R. Hasimoto-Beltran and A.A. Khokhar, "Spatial error concealment based on bezier curves," in *IEEE ICME 2005.*, 2005, pp. 996 –999.

[9] http://iphome.hhi.de/suehring/tml/, *Suehring JM17H.264/AVC*.

[10] "ISO/IEC JTC1/SC29/WG11, Report on Experimental Framework for 3D Video Coding, Doc. N11631," Guangzhou, China, October 2010.

[11] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *CSVT, IEEE Transactions on*, vol. 13, no. 7, pp. 560 –576, July 2003.

[12] L. Do, S. Zinger, and P.H.N. de With, "Objective quality analysis for free-viewpoint DIBR," in *ICIP, 2010*, 2010, pp. 2629 –2632.