



This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.

  
C O M M O N S D E E D

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**



**Attribution.** You must attribute the work in the manner specified by the author or licensor.



**Noncommercial.** You may not use this work for commercial purposes.



**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# A neighbourhood analysis based technique for real-time error concealment in H.264 intra pictures

Steven T.C. Beesley\*<sup>a</sup>, Christos Grecos<sup>a</sup>, Eran Edirisinghe<sup>b</sup>

<sup>a</sup>Electronic and Electrical Engineering, <sup>b</sup>Department of Computer Science,  
Loughborough University, Leicestershire, LE11 3TU, United Kingdom

## ABSTRACT

H.264s extensive use of context-based adaptive binary arithmetic or variable length coding makes streams highly susceptible to channel errors, a common occurrence over networks such as those used by mobile devices. Even a single bit error will cause a decoder to discard all stream data up to the next fixed length resynchronisation point, the worst scenario is that an entire slice is lost. In cases where retransmission and forward error concealment are not possible, a decoder should conceal any erroneous data in order to minimise the impact on the viewer. Stream errors can often be spotted early in the decode cycle of a macroblock which if aborted can provide unused processor cycles, these can instead be used to conceal errors at minimal cost, even as part of a real time system. This paper demonstrates a technique that utilises Sobel convolution kernels to quickly analyse the neighbourhood surrounding erroneous macroblocks before performing a weighted multi-directional interpolation. This generates significantly improved statistical (PSNR) and visual (IEEE structural similarity) results when compared to the commonly used weighted pixel value averaging. Furthermore it is also computationally scalable, both during analysis and concealment, achieving maximum performance from the spare processing power available.

**Keywords:** H.264, AVC, real-time, multimedia, error concealment, Sobel, edge detection

## 1. INTRODUCTION

MPEG-4 Part 10 Advanced Video Coding (AVC) / H.264 is the latest video coding standard from the Joint Video Team (JVT), a collaboration between the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) [1]. The standard has been designed to be suitable for a wide range of applications including storage and broadcast plus 2-way communications such as video telephony. It has also been designed with enhanced compression in mind such that it achieves considerably improved rate-distortion efficiency over existing standards such as MPEG-2 and H.263 [2].

A number of new coding features were incorporated in order to achieve this high level of compression, allowing up to 32 reference pictures for prediction, quarter pixel precision for motion compensation, variable block sizes including exact match integer transforms at 4x4 and 16x16, logarithmic quantisation step sizes and an in-loop deblocking filter to name just a few. However it is the new entropy encoding design that is of most interest to this paper, H.264 is structured such that all of the data below slice headers utilizes either Context-Adaptive Binary Arithmetic Coding (CABAC) or the lower complexity Context-Adaptive Variable Length Coding (CAVLC) compression techniques [3].

Stream errors may occur for a number of reasons, for example noise on a transmission line, hard drive corruption or scratches on an optical disc, alternatively excessive line congestion may cause unacceptable delays to a real time system. The very nature of entropy encoding makes a stream extremely vulnerable to such errors; a single bit error can render the entropy data useless until the stream can be resynchronised. In the case of H.264 this resynchronisation point will be the next fixed length coded pattern, for example the next data partition or slice header, therefore severe degradation can occur within an erroneous slice [4]. Several options exist to counteract such damage such as Forward Error Correction (FEC) which adds redundant information to the stream and Automatic Retransmission reQuest (ARQ) which allows erroneous data to be requested again, however both of these require additional bandwidth and introduce additional latency [5].

\*s.t.c.beesley@lboro.ac.uk; phone +44 (0) 1509 227083; fax +44 (0) 1509 227014; www.lboro.ac.uk

Multimedia on Mobile Devices 2007, edited by Reiner Creutzburg, Jarmo Takala, Jianfei Cai,  
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6507, 650706,  
© 2007 SPIE-IS&T · 0277-786X/07/\$18

SPIE-IS&T/ Vol. 6507 650706-1

There are times when adding latency to a stream is unacceptable (for example video telephony applications where a fluent transmission is more important than exact reconstruction of the data), bandwidth unavailable or communication only possible in one direction. In these cases the aforementioned resiliency tools may be unavailable and so a decoder has to rely on alternative flexibilities plus error concealment methods. The H.264 standard [3] also adds the option to use Flexible macroblock ordering (FMO); this allows a picture to be partitioned into multiple slices allowing for the creation of macroblock patterns that are better suited for error concealment. For example by splitting a picture into two slices in a checkerboard pattern, even if an entire slice is lost the damaged macroblocks will still have their horizontal and vertical neighbours from which to conceal from, using a half checkerboard pattern also provides diagonal neighbours.

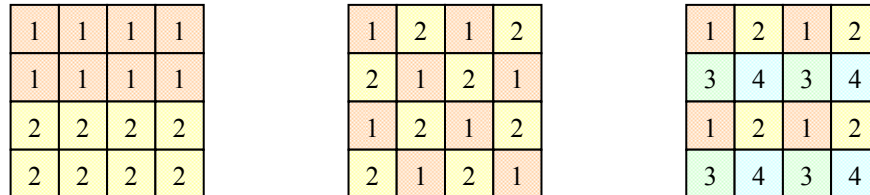


Fig. 1. FMO Examples, left: normal half split, centre: checkerboard, right: half checkerboard

Detection of errors is achieved by checking that control codes are valid and that video semantics are correct (for example that the number of macroblock coefficients received matches the expected number based on the macroblock type). In order to conceal erroneous data, damaged macroblocks are discarded and once the entire picture is decoded are then initially estimated from correctly received data in order to try and hide the visual repercussions of errors from the end user. Should less than 2 neighbouring blocks contain correctly decoded data then previously concealed macroblocks will also be used, as concealing is only an approximation it is important to provide as many neighbouring macroblocks as possible [6].

## 2. PROPOSED SCHEME

This paper concentrates on the concealment of intra slice macroblocks suitable for real time applications such as video telephony on mobile devices. No specific concealment techniques have being standardized in H.264 and so often only the minimal possible concealment is performed, for an intra picture this is weighted pixel value averaging, this is simply an average of linear interpolations in both horizontal and vertical directions and so is often referred to as bidirectional linear interpolation, or bilinear for short [7]. Whilst bilinear is the simplest and therefore quickest possible form of concealment it has two major shortfalls, it cannot conceal information outside of the horizontal and vertical planes and it assumes an equal weighting for both directions regardless of surrounding content, this means that the resulting reconstruction often looks overly smooth, blurred and smudged.

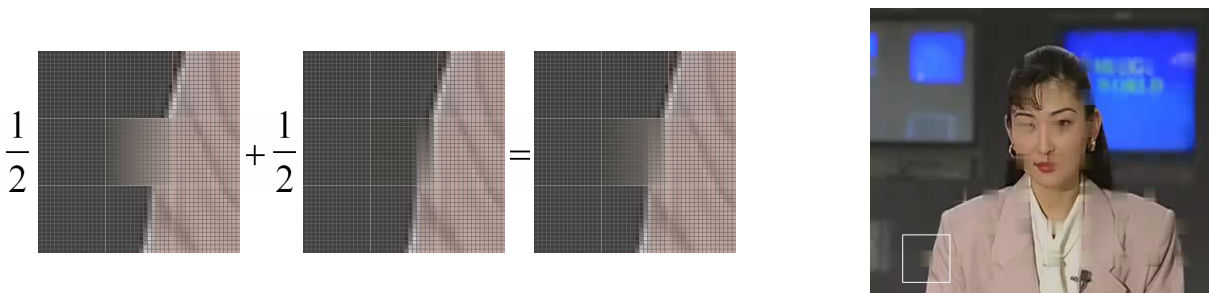


Fig. 2. Bilinear concealment example, the foreground gets smeared across the background

Previous work in [8] showed that by observing directional trends in the neighbouring pixels that the interpolations could be weighted off average giving rise to a significant improvement in both PSNR and the Structural SIMilarity (SSIM) visual metric described in [9] and [10], especially in cases where the macroblocks contained horizontal or vertical edges in which case the stronger direction was given a heavier weighting. The technique did however, not perform well where the macroblocks contained diagonal edges, extrapolating the original idea to a multi-directional approach quickly became computationally expensive and the lack of detailed edge identification meant that the technique was compromised; too few directions and diagonal edges were still lost, too many directions and even strong edges became blurred towards the centre of a macroblock.

The work presented in this paper has the same objective of using weighted interpolations in many directions in order to maintain diagonal edges; however it uses a different approach to the previous work. First edge convolution kernels such as Sobel shown in (1) are applied across pixels in the neighbouring macroblocks creating two gradient magnitudes per pixel,  $G_x$  and  $G_y$ .

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1)$$

These gradients are then used as the x and y components of a gradient vector making it simple to calculate the edge magnitude,  $G$ , and the edge direction,  $\theta$ , for each pixel

$$G = \sqrt{G_x^2 + G_y^2}, \theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (2)$$

Outputs of this for selected frames of different video sequences using the Sobel convolution kernels are shown in figure 3. Other kernels such as Prewitt and Roberts could also just of easily have been used.

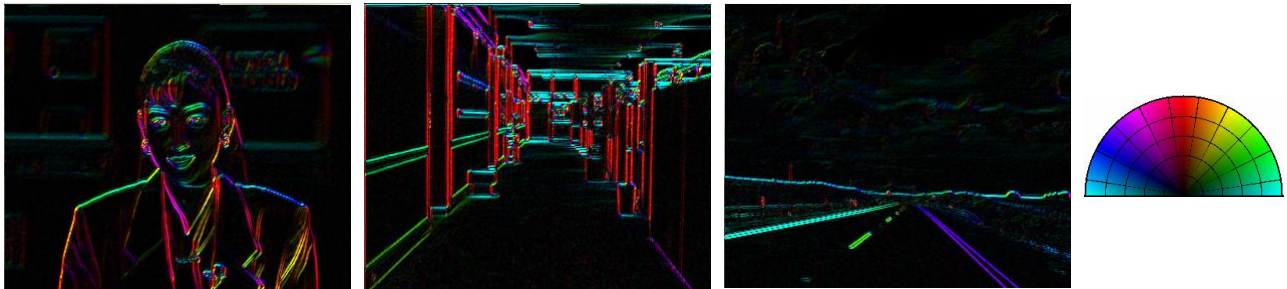


Fig. 3. Edge maps – Colour indicates direction, brightness indicates magnitude, as shown right

The edge information from pixels within macroblocks that neighbour erroneous blocks is then fed to the algorithm so that a weighted multi-directional interpolation can take place. In order to restrain the complexity of the algorithm, only a discrete number of directions are used for interpolation, the more directions are chosen the more complex the algorithm ends up but the better the results are. The choice of how many directions to use can therefore be made based on the quality required or the processing power available to the task, results for a differing number of directions are shown in section 3.

The first stage of the process is to quantise the edge directions to match an interpolation direction; if the number of steps is known prior to edge detection then this can automatically be incorporated into the inverse tangent lookup in order to further reduce the amount of required processing. The magnitudes of all pixels are then grouped by their direction and each group summed to give an overall magnitude for each direction ( $G_d$ ). These are then divided by the sum of all magnitudes ( $G$ ) to provide a weighting factor for each interpolation direction.

Due to the limited number of interpolation directions, concealment of a pixel in an erroneous macroblock is dependent on a relatively small number of bordering pixels pairs. The algorithm checks that these are in regions of valid data before calculating a pixel value (P) based on the sum of pixels linear interpolation value (I) multiplied by the weighting for each direction (d) as shown in (3).

$$P = \sum_{d=1}^n \left( I_d \times \frac{\sum G_d}{\sum G} \right) \quad (3)$$

An example of this which utilises four directions is shown in figure 4, in this case a checkerboard pattern FMO has been used and an entire slice lost, the consequence of which is that the diagonal neighbours are not available. For the highlighted pixel this means that only three directions have bordering pixels that reside in valid data and so the end value is a the weighted interpolations in the other three directions divided by the sum of magnitudes for the same three.

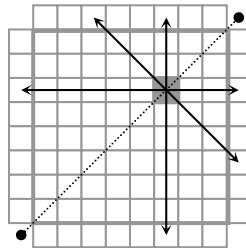


Fig. 4. Multi-directional interpolation example (4 directions)

### 3. EXPERIMENTAL RESULTS

Experimental results were collected using the JM10.2 reference H.264 decoder over multiple streams that had previously been encoded using main profile settings with two differing FMO patterns, a checkerboard pattern that utilized two slices and a half checkerboard pattern that utilized four slices (see figure 1). Whenever an intra picture was decoded an entire slice was removed giving error rates of 25% and 50%, the proposed scheme was then used over a range of directions in order to conceal the missing macroblock data, the results of which are shown in tables 1 to 4.

The tables show PSNR improvements over bilinear of up to 1.38dB and up to 11.24% improvements in SSIM (Akiyo, 50% data loss, 32 directions) and average improvements using 16 directions of 0.603dB and 5.03% over all sequences at both loss rates. They also show that only six directions need to be computed before improvements over bilinear concealment are made and that better gains are made when more directions are used, however there is a strong case of diminishing gains where eventually the increase in complexity may not justify the improvements made.

Individual frames for the Foreman, Highway, Hall and Container sequences are shown in figures 5 to 8 respectively. It can be seen that bilinear interpolation generally does a poor job of concealing the lost macroblocks that contain edges in all sequences; by contrast the edge preserving spatial correlation method performs extremely well for all but the few macroblocks where multiple edges intersect the lost macroblock or its neighbours. Using alternative convolution kernels produced results that were extremely close to those shown, but none managed to better the results using Sobel.

An un-optimised version of the concealment technique was also executed through the Intel® VTune Performance Analyzer to give a comparative indication of performance for a given number of directions. Results from this are shown in table 5 and figure 9, these show the total time spent inside the concealment function whilst decoding an entire sequence.

Table 1: PSNR after concealment, 25% loss (dB)

Dirs	Akiyo	Coastgu	Containr	Football	Foreman	Hall	Highway	Mot_dau	News	Silent
2	37.08	38.44	36.34	33.41	37.50	36.58	36.79	40.38	35.15	36.38
4	37.25	38.52	36.51	33.68	38.41	36.60	37.26	40.56	35.33	36.74
6	37.41	38.59	36.65	33.77	38.58	36.63	37.21	40.64	35.47	36.87
8	37.50	38.67	36.75	33.79	38.72	36.66	37.45	40.64	35.53	36.91
10	37.51	38.70	36.78	33.80	38.70	36.67	37.54	40.65	35.58	36.94
12	37.53	38.72	36.79	33.81	38.76	36.68	37.67	40.66	35.60	36.95
14	37.54	38.77	36.84	33.82	38.79	36.69	37.80	40.66	35.62	36.96
16	37.54	38.79	36.84	33.82	38.80	36.69	37.79	40.66	35.62	36.97
20	37.53	38.83	36.86	33.83	38.80	36.70	37.77	40.67	35.64	36.99
24	37.54	38.83	36.86	33.82	38.82	36.70	37.79	40.67	35.64	36.99
28	37.54	38.86	36.89	33.83	38.82	36.71	37.82	40.67	35.64	37.00
32	37.53	38.87	36.89	33.83	38.82	36.71	37.83	40.67	35.65	37.00
Bilinear	36.77	38.46	36.36	33.61	37.76	36.33	36.86	40.31	35.31	36.56

Table 2: SSIM after concealment, 25% loss (%)

Dirs	Akiyo	Coastgu	Containr	Football	Foreman	Hall	Highway	Mot_dau	News	Silent
2	60.08	37.43	43.07	33.43	42.06	52.11	50.04	54.97	45.93	35.22
4	62.11	37.2	42.82	34.71	50.01	51.86	52.29	56.47	46.13	38.51
6	62.86	37.45	43.46	35.31	51.28	51.86	52.57	57.15	46.99	39.61
8	63.67	38.18	44.08	35.61	52.44	52.20	53.75	57.22	47.42	40.00
10	63.89	38.52	44.29	35.78	52.30	52.22	54.29	57.34	47.74	40.36
12	64.00	38.66	44.4	35.83	52.70	52.19	54.83	57.42	47.91	40.50
14	64.13	39.42	44.81	36.00	52.91	52.33	55.32	57.37	48.09	40.56
16	64.12	39.54	44.91	35.98	53.02	52.33	55.29	57.43	48.18	40.64
20	64.02	40.08	45.16	36.04	53.05	52.47	55.21	57.48	48.37	40.77
24	64.15	40.07	45.16	36.03	53.17	52.41	55.33	57.48	48.40	40.81
28	64.22	40.55	45.36	36.12	53.21	52.54	55.43	57.47	48.47	40.85
32	64.13	40.58	45.37	36.10	53.22	52.52	55.44	57.50	48.50	40.88
Bilinear	57.80	34.89	43.48	34.45	43.82	48.40	50.59	54.87	45.55	36.82

Table 3: PSNR after concealment, 50% loss (dB)

Dirs	Akiyo	Coastgu	Containr	Football	Foreman	Hall	Highway	Mot_dau	News	Silent
2	32.57	37.61	34.04	31.61	35.99	35.15	36.15	39.1	33.33	34.97
4	35.49	38.14	35.01	31.69	36.17	35.38	36.47	39.18	33.60	35.18
6	35.56	37.96	34.83	31.84	36.62	35.35	36.42	39.33	33.65	35.36
8	35.83	38.00	35.04	32.00	36.76	35.39	36.52	39.39	33.76	35.49
10	35.87	38.01	35.05	32.02	36.83	35.41	36.62	39.42	33.78	35.52
12	35.93	38.03	35.11	32.06	36.86	35.43	36.80	39.44	33.86	35.55
14	35.97	38.08	35.20	32.10	36.89	35.45	36.87	39.46	33.91	35.56
16	36.00	38.10	35.22	32.12	36.91	35.45	36.93	39.47	33.94	35.58
20	36.01	38.15	35.24	32.11	36.92	35.47	36.91	39.48	33.97	35.59
24	36.04	38.17	35.26	32.14	36.94	35.48	36.94	39.49	33.98	35.61
28	36.04	38.19	35.30	32.14	36.95	35.49	36.95	39.50	34.00	35.61
32	36.04	38.21	35.31	32.14	36.95	35.49	36.96	39.51	34.01	35.61
Bilinear	34.66	37.53	34.27	31.82	36.16	35.00	36.17	38.97	33.16	35.12

Table 4: SSIM after concealment, 50% loss (%)

Dirs	Akiyo	Coastgu	Containr	Football	Foreman	Hall	Highway	Mot_dau	News	Silent
2	45.18	22.04	25.57	16.19	25.82	36.96	42.16	38.90	27.41	19.40
4	46.04	28.4	30.4	17.04	28.01	38.62	45.36	39.63	29.08	21.12
6	46.96	25.31	28.62	17.62	32.52	38.30	45.03	41.64	29.15	22.68
8	48.93	25.4	29.64	18.54	34.55	38.61	46.08	42.60	30.10	24.11
10	49.58	25.3	29.57	18.81	35.38	39.09	46.75	42.96	30.39	24.45
12	50.23	25.44	29.79	19.06	35.75	39.15	47.95	43.31	31.00	24.89
14	50.50	26.29	30.48	19.33	36.16	39.28	48.30	43.60	31.24	25.07
16	50.69	26.46	30.72	19.42	36.51	39.42	48.67	43.72	31.53	25.23
20	50.89	27.17	31.22	19.54	36.70	39.61	48.62	43.87	31.75	25.39
24	51.16	27.19	31.36	19.66	36.96	39.63	48.80	43.98	31.93	25.54
28	51.24	27.75	31.78	19.79	37.02	39.79	48.92	44.08	32.05	25.60
32	51.24	27.84	31.88	19.77	37.10	39.80	48.97	44.14	32.06	25.63
Bilinear	40.00	18.34	27.74	17.26	27.92	34.16	42.14	38.91	25.29	20.84



Fig. 5. Foreman frame 1 (I picture). 25% data loss

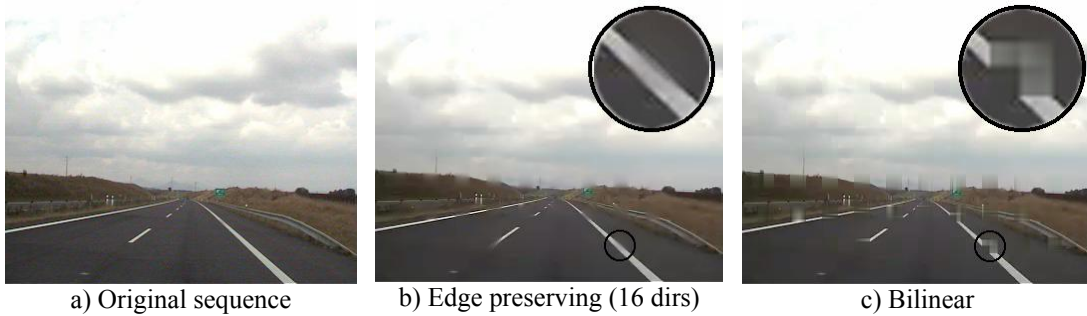


Fig. 6. Highway frame 18 (I picture). 25% data loss



Fig. 7. Hall frame 1 (I picture). 50% data loss

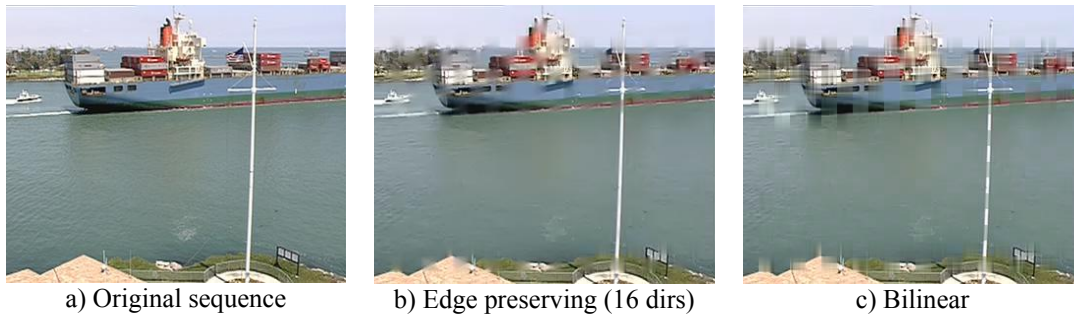


Fig. 8. Container frame 1 (I picture). 50% data loss

Table 5: Time to run concealment function

Dirs	Self Time ( $\mu$ S)	Total Time ( $\mu$ S)
2	1775472	4269295
4	2495060	6460558
6	3434803	9262870
8	4242627	11915847
10	5305534	15229639
12	6330309	18891710
14	7917575	23718849
16	9333199	29013462
20	12011627	38563638
24	15933353	50264268
28	19729110	65101716
32	24648937	82931737

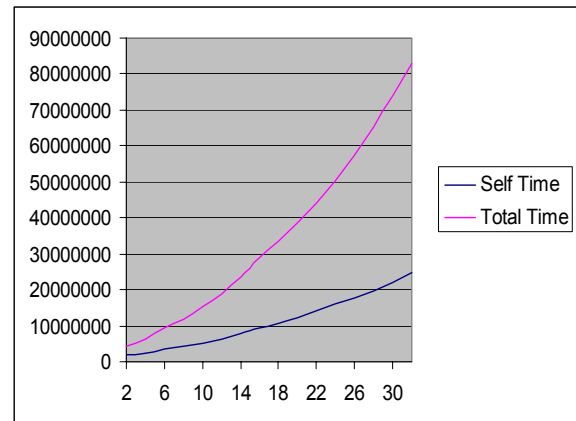


Fig. 9. Relative time comparison per direction

#### 4. CONCLUSIONS

The experimental results demonstrate significant mathematical and visual improvements of using a weighted, multi-directional interpolation technique over the commonly used bilinear interpolation, even when high error rate conditions of 50% were used. The failure of previous work indicates that the key to the successful improvements was in the interpolation weightings.

The technique demonstrated in this paper is deliberately kept computationally inexpensive; the early stages of the edge detection require only additions, subtractions and bit shifts, later calculations are simple and no additional processing such as noise removal or other pre-processing filters are needed. This coupled with the fact that the weightings only need to be calculated once per erroneous macroblock means that the use of Sobel convolution kernels adds little in the way of computational overhead to a multi-directional linear interpolation technique and yet have been proved to be extremely effective.

The algorithm is content independent in that it performs exactly the same number of calculations no matter what information the neighbouring blocks contain, it is also scalable in that the time needed to perform concealment in any given number of directions is known in advance and can be matched to the available number of spare cycles.



## REFERENCES

1. T. Wiegand, G.J. Sullivan, G. Bjøntegaard and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 13, 560-576 (2003).
2. T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards", *IEEE Transactions on Circuits & Systems for Video Technology*, 13(7), (2003).
3. Joint Video Specification, ITU-T Rec. H.264 | ISO/IEC 14496-10 (2002)
4. A. Raman and M. Babu, "A Low Complexity Error Concealment Scheme for MPEG-4 Coded Video Sequences", *IEEE Symposium on Multimedia Communications and Signal Processing*, Bangalore, India. (2001).
5. F. Chiaraluce, L. Ciccarelli, E. Gambi and S. Spinsante, "Performance Evaluation of Error Concealment Techniques in H.264 Video Coding", Picture Coding Symposium, San Francisco, CA, USA (2004)
6. O. Nemethova, A. Al-Moghrabi and M. Rupp, "Flexible Error Concealment for H.264 Based on Directional Interpolation", *Proc. IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, 2, 1255-1260, (2005).
7. Y.K. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta and M. Gabouj, "The Error Concealment Feature in the H.26L Test Model." *Proc. IEEE International Conference on Image Processing*, 2, 729-732, (2002).
8. A.J. Armstrong, S.T.C. Beesley, C. Grecos and D.J. Parish, "Directionally Sensitive Bilinear Concealment for H.264", *Proc. Visualization Imaging and Image Processing*, Benidorm, Spain, (2005).
9. Z. Wang, L. Lu and A. Bovik, "Video Quality Assessment Based On Structural Distortion Measurement", *SP:IC*, 19-2, 121-132, (2004).
10. Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity", *IEEE Trans. Im. Proc.*, 13-4, 600-612, (2004).