

# A Spatially Enhanced Error Concealment Technique and its Potential Alternative Application to Reduce H.264 Stream Sizes

Steven Beesley and Christos Grecos

Original scientific paper

**Abstract** - With more and more video content being transmitted digitally and with user expectations continually rising, error concealment is becoming an increasingly important part of streaming media. Often overlooked in the past, even now manufacturers are often only doing the bare minimum necessary in order to avoid complexity. This paper first presents a combination of simple techniques that when combined produce an extremely effective concealment method that maintains spatially correlated edges throughout any lost data; this in turn gives an increase in both mathematical and visual performance when compared against the commonly used bilinear concealment technique. Secondly this paper looks at an alternative use of the bilinear passive error concealment algorithm that is often used by H.264 decoders. Occasionally a concealed macroblock is mathematically closer to the original than an encoded and decoded one, by removing these from the stream at the encoder and thus forcing the decoder to conceal the missing data, a significant reduction in the bit stream size (up to 5%) can be achieved with almost no loss in quality.

**Index terms** - Error Concealment, Spatial, Sobel, Rate Reduction, Edge Detection

## I. INTRODUCTION

H.264 / MPEG-4 Part 10 Advanced Video Coding (AVC) is the latest video coding standard from the Joint Video Team (JVT), a collaboration between the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG) [1]. The standard has been designed to be suitable for a wide range of applications including 2-way communications such as video telephony plus 1-way streams such as for storage and broadcast purposes. 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].

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 are 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.

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].

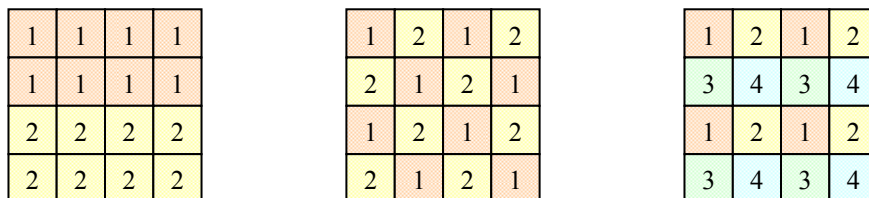


Fig. 1. FMO Examples, left: normal half split, centre: checkerboard, right: half checkerboard

II. ENHANCED SIMPLE ERROR CONCEALMENT METHOD

The first part of this paper concentrates on the concealment of intra slice macroblocks for which no specific concealment techniques have been standardized in H264. Typically, only the minimal possible concealment is performed for an intra picture based on the 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]. Bilinear 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.

directions 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 all available pixels in the neighbouring macroblocks creating two gradient magnitudes per pixel,  $G_x$  and  $G_y$ .

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

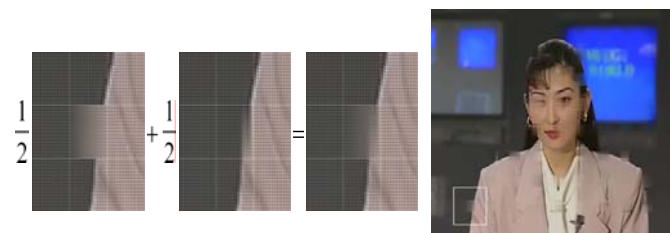


Fig. 2. Bilinear concealment example, the edge gets smeared

Previous work in [8] showed that by observing directional trends in the neighbouring pixels, the interpolations could be weighted off thus giving rise to a significant improvement in both PSNR and the IEEE Structural SIMilarity (SSIM) visual metric described in [9] and [10]. This improvement was bigger when macroblocks contained horizontal or vertical edges of unequal strength, in which case the stronger direction was given a heavier weighting. The technique did however not perform well where the macroblocks contained diagonal edges since 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. In particular, diagonal edges were still lost due to too few directions, whereas in the case of too many

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,  $q$ , for each pixel

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

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 as easily have been used.

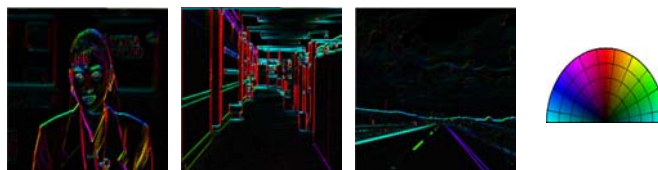


Fig. 3. Edge maps – Colour indicates direction, brightness indicates magnitude, as shown right

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, thus the more directions chosen the more complex the algorithm is and 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 and is dependent on the number of directions as 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 that utilises four directions is shown in figure 4. In this case, a checkerboard pattern FMO has been used and an entire slice lost, thus 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 the weighted interpolation in the other three directions.

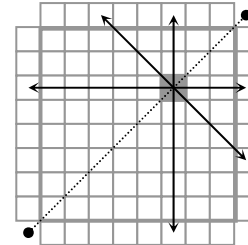


Fig. 4. Multi-directional interpolation example (4 directions)

### III. ERROR CONCEALMENT 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. We used 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 and 2.

TABLE 1  
PSNR AFTER CONCEALMENT, 25% LOSS (DB)

Dirs	Akiyo	Coast	Conta	Fball	Fman	Hall	Hway	Mo_da	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
Bil	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	Coast	Conta	Fball	Fman	Hall	Hway	Mo da	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
BIL	57.80	34.89	43.48	34.45	43.82	48.40	50.59	54.87	45.55	36.82

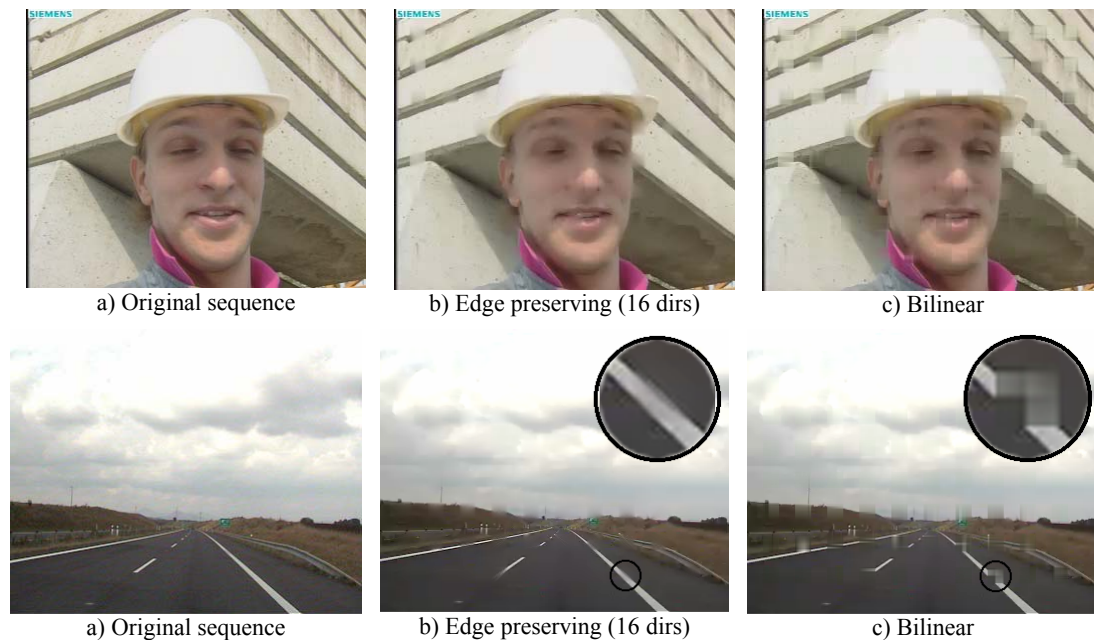


Fig. 5. Top - Foreman frame 1 (I). 25% data loss, Bottom - Highway frame 18 (I). 25% data loss

The tables show PSNR improvements over bilinear of up to 1.06dB and up to 9.4% improvements in SSIM (Foreman, 32 directions) and average improvements using 16 directions of 0.603dB and 5.03% over all sequences. 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 and Highway sequences are shown in figure 5. It can be seen that bilinear interpolation generally does a poor job of concealing the lost macroblocks that contain edges in both 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.

#### IV. RATE REDUCTION USING ERROR CONCEALMENT

The second part of this paper looks at an alternative use of the passive error concealment algorithm where over a reliable but bandwidth limited network the bit stream size can be further reduced. As each macroblock is encoded, it is first passed through one of several transform functions and is then quantised, the latter of which can introduce significant errors depending on the Quantisation Parameter (QP) setting. In some cases the reconstruction errors are so large that concealing the macroblock from its encoded neighbours gives a closer mathematical match to the original source data than can be achieved by the normal (error free) decoding process.

The scheme operates as follows. During the encoding process of a H.264 stream, every macroblock of every picture is also concealed from its neighbours using weighted pixel value averaging (bilinear), regardless of slice type. The PSNR of every encoded macroblock after local reconstruction is calculated against the original source file along with the PSNR of the concealed version. If the concealed version improves the PSNR, then sending the macroblock data over the channel unnecessarily increases the size of the bit stream. Within this section it is assumed that the macroblock can simply be removed from the stream, however due to the way the stream is constructed this is not actually possible. Section 5 looks in more depth at why this is the case and at some of the available possibilities for circumventing this problem.

There are several reasons why not all improved macroblock versions should be removed from the stream, for example should all marked macroblocks be removed then there could be occurrences of horizontally or vertically neighbouring macroblocks being deleted. As the concealed PSNR is predicted with the assumption that all neighbouring blocks are available then this would lead to unpredictable results. This uncertainty could be avoided by recalculating the concealed PSNR at every step; however this would add an additional and significant computational overhead to the technique.

As an alternative, the marked macroblocks are all added to a list which is ordered by bit stream size so that the macroblocks with the largest potential saving are given the highest priority. This order was chosen to reduce the rate of the bit stream; however a different order could easily be used, for example where a PSNR improvement or a better RD is considered a more advantageous result.

The encoder then runs through this list starting with the highest priority, if all of the neighbours are still available then the macroblock can be removed from the sequence, however if any of the neighbours have already been marked for removal then the macroblock is ignored on the assumption that a concealed version cannot be accurately reconstructed.

A more aggressive approach can be taken using this scheme where once macroblocks with an average PSNR improvement are removed, the process is repeated on just the chrominance channels. As the chrominance channels are coarser in resolution an increased number of macroblocks are marked for removal, however they also contain less data and therefore less potential savings.

## V. RATE REDUCTION OVERHEADS

When decoding a H.264 stream, macroblock positions are calculated from the slice header (which gives the first macroblock number in the slice), the number of macroblocks that have been decoded since this header and any slice pattern information (for example FMO). For this reason, it is not possible to simply remove macroblocks from the stream.

Whilst a decoder may be able to work out how many macroblocks it is missing from each slice, it would not be able to work out where in the slice they are missing from. This section looks at several ways that the positional information can be sent to the decoder.

Flexible Macroblock Ordering (FMO) may initially seem like an ideal solution as it should allow the blocks flagged for entire deletion to be grouped into a single slice. For example the map type 6 which allows explicit assignment, enables a slice to be ignored so that it is never encoded into the stream. A decoder should be able to parse the remainder of the picture and flag the slice and relevant macroblocks as missing and therefore erroneous. Unfortunately, using this particular map type means that every remaining macroblock number gets added to a list inside one of the slice headers thus increasing the bit stream size beyond that of normal savings and especially at low QP settings where very few blocks are removed. It is therefore not suitable for bit stream reduction.

Another option is to tell the decoder exactly what is going on with each macroblock. Only a single bit per block is needed to inform the decoder whether to conceal or not and if the block is set to be concealed then another bit can be sent to tell the decoder whether to conceal the entire block or just the chrominance channels. Should this information be sent as is, then it would normally increase the bit stream size. However, using lossless compression such as Run Length Encoding (RLE) allows for some limited savings when compared to the maximum originally predicted. This data needs to be protected to ensure it reaches the decoder as without it the stream cannot be rendered properly and the resulting bit stream would not be standard compliant.

The final option considered is to generate a new macroblock type for each type of slice; again compliant decoders will be unable to render such a stream, however this solution is much closer to a standard approach [11]. For the purpose of our results, the binarization of the following macroblock types have been extended to provide the extra modes needed:

- I\_PCM mode extended from 11 to 110 to create I\_Conceal 1110 and I\_Conceal\_Cr 1111.
- P\_8x8 mode extended from 001 to 0010 to create P\_Conceal 00110 and P\_Conceal\_Cr 00111.
- B\_Direct\_16x16 mode extended from 0 to 00 to create B\_Conceal 010 and B\_Conceal\_Cr 011.

These were selected as they were the least used over the tested sequences. In the absence of sequence specific knowledge it is possible though that such a scheme could give an increase in stream size. In order to alleviate this problem, an adaptive method could be used where both the encoder and decoder keep count of how many times each macroblock has been used. In this manner, both would then be aware of the least used macroblock type which could be used for concealment purposes. Evidently though this scheme would make the stream less robust as any transmission errors would cause the decoder to lose synchronisation with the encoder.

VI. RATE REDUCTION EXPERIMENTAL RESULTS

Experimental results have been gained using the JM10.2 encoder reference software over several 4:2:0 YUV sequences chosen for their differing characteristics, namely Akiyo, Coastguard, Football and Foreman. Results were calculated for both CIF and QCIF versions using 101 frames with main profile settings (77 with an I-B-P-B-P-B group of pictures) and 100 frames using baseline profile settings (66 with a single I slice followed by all P slices). Each tested sequence was encoded several times over a series of quantisation parameters ranging from 28 (high quality) to 40 (low quality).

In all cases, other settings were left as in the standard and rate control was disabled; both PSNR and SSIM [9,10] visual metrics were recorded.

Tables 3 and 4 show a summary of the results collected for each sequence, profile and QP setting. The maximum bit stream saving rows indicate the percentage reduction in bit stream size if the macroblocks were simply discarded. This is the maximum saving possible as it does not take into account any necessary overheads. Also present are  $\Delta$ PSNR and  $\Delta$ SSIM which show the impact on video quality caused by concealing the chosen blocks. In both cases a positive number represents an improvement in quality.

TABLE 3  
CIF SEQUENCE RATE REDUCTION RESULTS

Profile	Main				Baseline			
	28	32	36	40	28	32	36	40
<b>Quantisation Parameter</b>								
<b>Sequence</b>	<b>Akiyo</b>							
<b>Max. Bit Stream Saving (%)</b>	0.91	1.62	4.17	4.58	0.17	0.59	2.00	3.78
<b>Max. RLE Saving (%)</b>	0.34	0.37	1.69	1.42	0.06	0.14	0.72	0.63
<b>New MB Type Saving (%)</b>	0.59	0.89	2.62	2.60	0.12	0.40	0.53	2.37
<b><math>\Delta</math>PSNR (dB)</b>	0.020	0.017	0.020	0.040	0.018	0.095	0.100	0.124
<b><math>\Delta</math>SSIM (%)</b>	-0.01	-0.12	-0.57	-0.51	0.01	0.01	0.05	-0.27
<b>Sequence</b>	<b>Coastguard</b>							
<b>Max. Bit Stream Saving (%)</b>	0.18	0.79	1.80	4.68	0.14	0.68	1.84	3.83
<b>Max. RLE Saving (%)</b>	0.09	0.48	0.95	1.81	0.09	0.45	1.10	1.81
<b>New MB Type Saving (%)</b>	0.13	0.65	1.45	3.51	0.11	0.56	1.46	2.80
<b><math>\Delta</math>PSNR (dB)</b>	0.000	-0.01	0.013	0.070	0.130	-0.05	-0.03	0.044
<b><math>\Delta</math>SSIM (%)</b>	-0.25	-0.79	-0.92	-0.9	-0.18	-0.62	-0.76	-0.37
<b>Sequence</b>	<b>Football</b>							
<b>Max. Bit Stream Saving (%)</b>	0.07	0.25	0.92	1.69	0.07	0.32	0.75	1.06
<b>Max. RLE Saving (%)</b>	0.00	0.07	0.34	0.51	0.01	0.13	0.32	0.39
<b>New MB Type Saving (%)</b>	0.03	0.16	0.65	1.09	0.03	0.22	0.50	0.68
<b><math>\Delta</math>PSNR (dB)</b>	0.003	0.000	-0.01	0.01	0.019	0.077	-0.04	0.064
<b><math>\Delta</math>SSIM (%)</b>	-0.01	-0.27	-0.93	-0.41	0.01	-0.46	-0.43	-0.13
<b>Sequence</b>	<b>Foreman</b>							
<b>Max. Bit Stream Saving (%)</b>	0.74	1.56	2.64	4.53	0.70	1.22	1.76	2.67
<b>Max. RLE Saving (%)</b>	0.17	0.43	0.76	1.52	0.24	0.42	0.60	0.82
<b>New MB Type Saving (%)</b>	0.46	0.96	1.58	2.77	0.46	0.81	1.18	1.74
<b><math>\Delta</math>PSNR (dB)</b>	0.000	0.013	0.033	0.050	-0.03	-0.06	0.274	-0.07
<b><math>\Delta</math>SSIM (%)</b>	-0.42	-0.68	-0.82	-0.62	-0.30	-0.25	-0.10	-0.09

When calculating the RLE results, 8 different methods of compression were tested including only adding run length data to non-concealed blocks. It was observed that it is unusual to get a run of more than one concealed block due to the neighbour restriction and that runs only occur at edges where the end of one row and the beginning of the next are both marked for concealment. Some of the RLE methods only

looked at I slices where most of the savings occur and some only looked at the entire macroblock removal thus ignoring the additional chrominance step. For the latter case however, the small additional savings are not always worth the extra overhead. The maximum RLE saving results show the savings from the method that worked best for each sequence.

TABLE 4  
QCIF SEQUENCE RATE REDUCTION RESULTS

Profile Quantisation Parameter	Main				Baseline			
	28	32	36	40	28	32	36	40
Sequence	<b>Akiyo</b>							
Max. Bit Stream Saving (%)	0.60	0.51	1.89	2.44	0.14	0.12	0.45	1.21
Max. RLE Saving (%)	0.26	0.09	0.76	1.07	0.07	0.02	0.23	0.74
New MB Type Saving (%)	0.43	0.28	1.30	1.64	0.10	0.05	0.33	0.98
$\Delta$ PSNR (dB)	-0.02	0.032	0.000	0.054	0.000	-0.01	0.123	-0.12
$\Delta$ SSIM (%)	-0.82	0.00	-0.54	-0.88	0.66	-0.04	-0.15	-0.57
Sequence	<b>Coastguard</b>							
Max. Bit Stream Saving (%)	0.25	1.75	1.93	5.24	0.22	2.14	3.88	2.96
Max. RLE Saving (%)	0.10	1.17	0.78	2.56	0.14	1.49	2.28	1.23
New MB Type Saving (%)	0.17	1.48	1.44	3.92	0.18	0.78	3.00	2.12
$\Delta$ PSNR (dB)	0.008	0.075	0.00	0.086	0.039	0.032	0.102	0.070
$\Delta$ SSIM (%)	-0.30	-1.44	-0.96	-0.58	-0.28	-1.18	-0.64	0.07
Sequence	<b>Football</b>							
Max. Bit Stream Saving (%)	0.05	0.30	1.05	0.88	0.09	2.14	3.88	0.48
Max. RLE Saving (%)	0.01	0.13	0.46	0.24	0.03	1.49	2.28	0.14
New MB Type Saving (%)	0.03	0.23	0.79	0.54	0.06	0.78	3.00	0.27
$\Delta$ PSNR (dB)	-0.03	0.065	0.044	-0.15	0.005	0.032	0.102	-0.03
$\Delta$ SSIM (%)	-0.11	-0.53	-0.84	-0.22	-0.28	-1.18	-0.64	-0.25
Sequence	<b>Foreman</b>							
Max. Bit Stream Saving (%)	0.26	0.57	0.70	1.65	0.20	0.33	0.28	0.83
Max. RLE Saving (%)	0.06	0.13	0.14	0.62	0.04	0.07	0.01	0.19
New MB Type Saving (%)	0.15	0.33	0.34	0.98	0.12	0.21	0.15	0.53
$\Delta$ PSNR (dB)	0.052	0.004	0.075	0.027	0.091	0.041	0.122	0.312
$\Delta$ SSIM (%)	-0.02	-0.17	-0.21	-0.36	-0.21	0.02	-0.10	-0.80

Figure 6 shows two sets of results, the first frame from the Akiyo sequence at QP28 and QP40 respectively. Macroblocks that are removed are highlighted in the middle picture where a white block represents chrominance channel removal and a

black-bordered box with diagonal line represents the entire block being deleted. The neighbour restrictions are clearly visible when QP40 is used where an emerging checkerboard pattern is more than apparent.



a) Normally decoded



b) Removed blocks highlighted



c) Blocks concealed



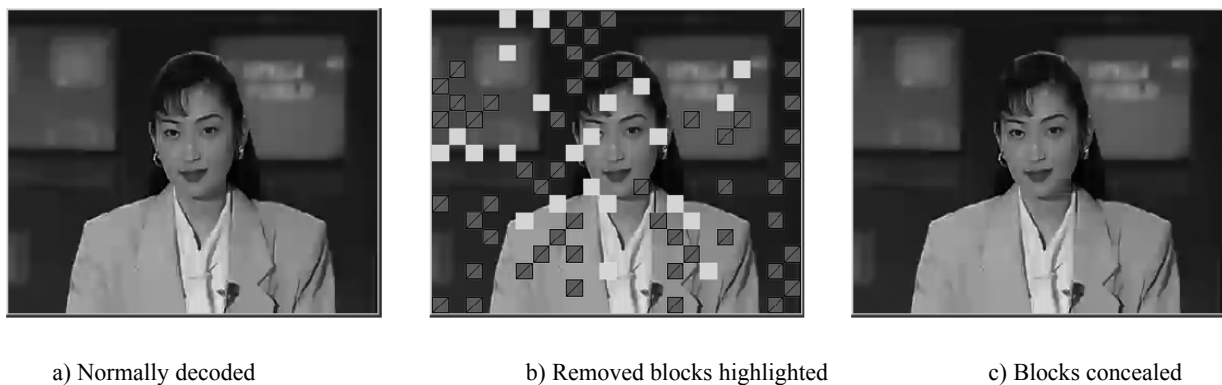


Fig. 6. Akiyo CIF Main profile, Frame 1 QP28 (top) and Frame 4 QP40 (bottom)

## VII. DISCUSSION AND CONCLUSIONS

It would be instructive to compare our scheme with others published in the literature. For example, our previous work in [8] had issues in the presence of many diagonal edges inside the lost macroblocks since extrapolating macroblock bordering pixels to a multi-directional approach quickly became computationally expensive and the lack of detailed edge identification meant that the technique was compromised. In particular, diagonal edges were still lost in the missing macroblocks if too few extrapolation directions were chosen, whereas in the case of too many directions even strong edges became blurred towards the centre of a macroblock. The work in [15] uses a different interpolation formula than ours and is slightly more computational intensive since it computes entropy plus directional weights as opposed to a fixed set of directional weights only in our scheme. That work does not consider visual implications as measured by the SSIM metric and the experiments assumed much less error prone channels than our scenarios (4% errors in theirs versus 25-50% errors in ours).

The error concealment experimental results of the proposed scheme demonstrate the 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 shortcomings of previous works clearly indicate that the key to the successful improvements was in the interpolation weightings. The technique is deliberately kept computationally inexpensive. The early stages of the edge detection require only additions, subtractions and bit shifts, while subsequent 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 our technique.

Regarding the rate reduction due to our scheme, two patterns emerge from testing. Firstly, the savings are relatively moderate for high quality video settings. Secondly, the lower the video quality due to an increase in quantisation step size,

the larger the potential saving. This is due to the quantisation effects of the reconstructed video being more severe - making weighted pixel value averaging superior in terms of PSNR gain. This in turn allows a larger number of macroblocks to be removed.

Of particular note are the foreman and akiyo sequences, both of which have a large number of smooth areas and are therefore ideally suited to the bilinear concealment technique. When the quantisation parameter is set to 40, both potentially achieve just under 5% reduction in bit stream size (2% after overheads) as well as improving the overall PSNR. This technique is less effective for high texture video streams such as "Football" when compared to the other sequence types due to the concealment technique being unable to recreate a better representation than the standard decoding cycle achieves.

Despite only removing macroblocks when concealment provides an immediate improvement, it can be seen that over an entire sequence there are both minor improvements and deteriorations in PSNR. This is due to the simplicity of the active selection, where the effects on slices that use the macroblock for prediction are not calculated. This could be implemented and would guarantee an overall improvement in PSNR, however would greatly increase the computational complexity of the algorithm. Conversely, for a given drop in PSNR even larger savings could be achieved.

Improving the visual aspect of error concealment is an ongoing area of research (sections 1-3), [12-14]. It is intended to utilise some of this work to improve the perceived playback quality when using this technique. Utilising an improved concealment method would also be likely to have the advantage of more macroblocks being selected for removal further increasing the reduction in bit stream for some video sequences.

The SSIM metric is designed to give an objective similarity measurement that better matches the human visual system. The results show a consistent but small reduction in this metric over all test cases implying that despite generally gaining improved PSNR results, the method is in fact marginally reducing the video quality. This could be overcome by also comparing the SSIM metric and only removing those



macroblocks with both an increased PSNR and SSIM result, however as the difference is only small and the increase in encoding time to do this would be significant, it was felt unnecessary.

One slight disadvantage of our technique is that it potentially makes the sequence less robust. Any macroblocks lost during transmission, in combination with one or more of their neighbouring macroblocks being removed by the encoder will give unpredictable concealment results. Whilst the likelihood of this is that any further concealment would give worse PSNR results, it is entirely sequence and video quality dependent. The reduction in robustness may be considered an acceptable trade-off for the savings generated, especially for lower quality video sequences. Certainly in a real-world streaming application, network conditions or storage reliability should be considered.

It is clear that this technique is ideally suited to situations where bandwidth is limited or costs are at a premium, such as found on mobile phone networks. On these networks any savings in bandwidth without major detriment to quality are useful in providing functionality at reasonable cost. Further modifications will make the technique useful under broader conditions such as satellite digital video.

#### 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] C.Yim, W.Kim and H.Lim, "Hybrid error concealment method for H264 video transmission over wireless networks", *International Conference on Communications and Mobile Computing*, Honolulu, Hawaii, USA, pp. 606-611 (2007).
- [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, pp. 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).
- [11] Alexis M. Tourapis, Feng Wu, Shipeng Li, "Direct macroblock coding for predictive (P) pictures in the H.264 standard", *SPIE Visual Communications and Image Processing (VCIP)*, 5308, 364-371, 2004
- [12] P. Nasiopoulos, L. Coria-Mendoza, H. Mansour, & A. Golikeri, "An Improved Error Concealment Algorithm for Intra-Frames in H.264/AVC", *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, 2005
- [13] A.Reibman, D.Poole, "Characterising packet loss impairments in compressed video", *IEEE International Conference in Image Processing*, 2007, Sept 16-19, San Antonio, TX, USA.
- [14] X.Feng, T.Liu, D.Yang and Y.Wang, "Saliency based objective quality assessment of decoded video affected by packet losses", *IEEE International Conference in Image Processing*, 2008, Oct 12-15, San Diego, CA, USA
- [15] Agrafiotis, D., Bull, D. R., and Canagarajah, N., "Spatial error concealment with edge related perceptual considerations", *Signal Processing: Image Communication*, Volume 21, Issue 2, pp. 130-142, February 2006.



**Christos Grecos** holds a Bachelor's in Computer Science (Systems Software Option) from the Concordia University, Montreal, Canada (1994), an MSc in Human Computer Interaction from the Heriott Watt University, Edinburgh, Scotland (1995) and a PhD in 2001 in Standard Compliant Image and Video Coding algorithms from Glamorgan University, UK. Currently, he is a Reader in Visual Communication Standards in the School of Computing, Engineering and Physical Sciences at the University of Central Lancashire, UK. Grecos has wide experience in the design and implementation of video compressors across the JPEG, MPEG and H264 families of standards and has published more than 70 papers in these areas. He is a Senior Member of *IEEE*, a Senior Member of *SPIE* and a Member of the UK MPEG committee.