Backward Compatible High Dynamic Range MPEG Video Compression

Rafał Mantiuk, Alexander Efremov, Karol Myszkowski, and Hans-Peter Seidel
MPI Informatik

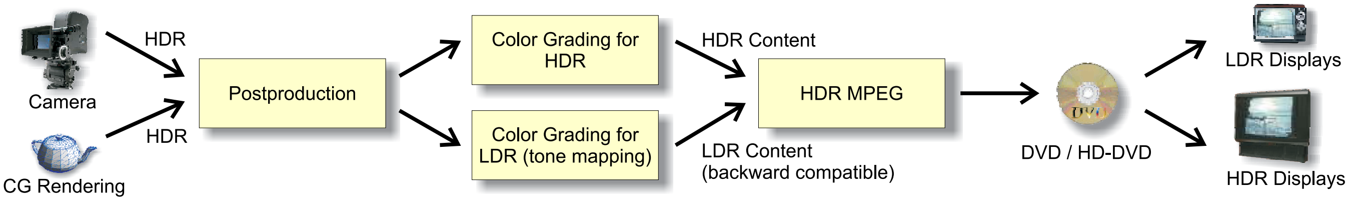

Figure 1: Our backward-compatible HDR DVD movie processing pipeline. The high dynamic range content, provided by advanced cameras and CG rendering, is encoded in addition to the low dynamic range (LDR) content in the video stream. The files compressed with the proposed HDR MPEG method can play on existing and future HDR displays.

Abstract

To embrace the imminent transition from traditional low-contrast video (LDR) content to superior high dynamic range (HDR) content, we propose a novel backward-compatible HDR video compression (HDR MPEG) method. We introduce a compact reconstruction function that is used to decompose an HDR video stream into a residual stream and a standard LDR stream, which can be played on existing MPEG decoders, such as DVD players. The reconstruction function is finely tuned to the content of each HDR frame to achieve strong decorrelation between the LDR and residual streams, which minimizes the amount of redundant information. The size of the residual stream is further reduced by removing invisible details prior to compression using our HDR-enabled filter, which models luminance adaptation, contrast sensitivity, and visual masking based on the HDR content. Designed especially for DVD movie distribution, our HDR MPEG compression method features low storage requirements for HDR content resulting in a 30% size increase to an LDR video sequence. The proposed compression method does not impose restrictions or modify the appearance of the LDR or HDR video. This is important for backward compatibility of the LDR stream with current DVD appearance, and also enables independent fine tuning, tone mapping, and color grading of both streams.

CR Categories: I.4.2 [Image Processing and Comp. Vision]: Compression (Coding)—Approximate methods

Keywords: video compression, high dynamic range, backward compatibility, MPEG, tone mapping, visual perception, prefiltering, visual masking, visible difference predictor

1 Introduction

There is tremendous progress in the development and accessibility of high dynamic range (HDR) imaging technology [Reinhard et al. 2005]. Modern graphics hardware pipelines are HDR-enabled and HDR cameras and display devices are available [Seetzen et al. 2004]. In concert with those developments, 3D graphics and image processing software often include HDR capabilities. The entertainment and gaming industries already employ HDR techniques. Movies digitized from celluloid film can easily capture a dynamic range of 4–5 orders of magnitude and modern digital cinema cameras, such as Dalsa's Origin® or Thomson Viper FilmStream$^{\mathrm{TM}}$, feature extended dynamic range capture. Adjusting the dynamic range to a specific display device through tone and gamut mapping can be performed in real time and added as a final processing stage to any digital video player, game engine, or real-time rendering system [Krawczyk et al. 2005].

Given such developments, an efficient method for exchanging HDR information among various HDR devices and software packages is required. Analogous to traditional (LDR) digital imaging technology, there is a need for *lossy* compression schemes for HDR images and video, similar to JPEG and MPEG standards. An important requirement for such schemes is backward compatibility with existing low dynamic range (LDR) technology. This problem has already been successfully addressed by Ward and Simmons [2004] for HDR JPEG encoding. In this work, we focus on backward-compatible HDR MPEG compression techniques, which guarantee that HDR video can be played on any modern DVD system and the resulting video appearance is the same as for traditional DVDs. The latter issue is of primary importance since the DVD industry puts much expertise and effort into tuning the video appearance on many typical LDR display devices (a difficult task given a single LDR video stream that is stored on DVDs) and would never accept any hazardous or intentional (third party) departure from the tuned appearance.

The greatest advantage of HDR video can be expected for HDR displays (e.g., BrightSide DR37-P), but even nominally LDR devices can often take the advantage of HDR information. The quality of existing DVD content is inadequate for many existing display devices such as high power projectors and 10–12-bit desktop LCD displays. The limited bit-depth per color channel leads to the loss of low contrast details and false contours in smooth gradient regions. Such artifacts become a problem for modern LCD television sets that feature sharper, brighter, and less noisy images than traditional CRT displays for which DVD was primarily developed [Daly and Feng 2004]. Clearly, a new HDR video encoding, which is fully compatible with existing LDR devices and at the same time enables for HDR-aware devices to take advantage of HDR information, is urgently needed.

In this work we propose a novel backward-compatible HDR MPEG video encoding (as depicted in Figure 1). The backward compatibility is achieved by encoding the HDR and LDR video frames in an LDR stream that is compatible with MPEG decoders, and a residual stream that enables the restoration of the original HDR stream. To minimize redundancy of information, the residual and LDR streams are decorrelated. Such decorrelation requires perceptually meaningful comparison of the LDR and HDR pixels, which we achieve by introducing a pair of corresponding color spaces that are scaled in terms of the human visual system (HVS) response to luminance and chrominance stimuli. We use these color spaces to build a frame-dependent reconstruction function that approximates values of HDR pixels based on their LDR counterparts. This approximation, together with other compression mechanisms, reduces the residual stream to about 30% of the size of the LDR stream. Because of such a small overhead, both standard resolution and High-Definition movies can fit in their original storage medium when encoded with HDR information. Since the proposed HDR MPEG encoding does not impose any restrictions on LDR or HDR content, both videos can be independently tuned and tone/gamut mapped to achieve the best look on different classes of displays. This tuning flexibility is required for current practices of the DVD industry. To reduce the production costs of HDR DVD players, the compression algorithm is designed so that standard 8-bit MPEG decoding chipsets can be used to decode the HDR stream.

A second important contribution of this work is the development of a perception-based HDR filter that predicts the visibility thresholds for HDR frames. Our wavelet-based filtering approach is fast as required by video applications, but still models important characteristics of the HVS such as luminance masking, contrast sensitivity, and visual masking for the full visible dynamic range of luminance. We apply our HDR filter to remove invisible noise in the residual video stream taking into account the adaptation conditions and visual masking imposed by the original HDR stream. This leads to even more effective HDR video compression since details that cannot be seen are removed from the residual stream prior to encoding.

2 Previous work

This work is the first attempt at developing a lossy, backward-compatible HDR video encoding with proper color handling. Since the problem of HDR lossy compression has already been partially addressed, especially for images, we briefly discuss existing solutions from the standpoint of their application to HDR video.

The problem of dynamic range compression and expansion arises in many imaging pipelines with constrained bit depth at certain processing stages (only 6 bits per color channel is often used for DVD movies while displays can handle 8 bits/color channel). This may result in the loss of low amplitude signals and false contouring. Bit-depth expansion (BDE) techniques are designed specifically to combat those effects and achieve higher perceived bit depth quality than are physically available. For example imperceptible spatiotemporal noise is added to an image prior to the quantization step in dither techniques [Daly and Feng 2003]. When higher bit depth information is not available, low-amplitude details cannot be reconstructed, and processing is focused on removing false contours using adaptive filtering, predictive cancellation, spatial frequency channel coring techniques [Daly and Feng 2004]. All existing BDE and de-contouring techniques are optimized for much lower bit depth expansion than required to accommodate HDR image and video content. Furthermore, storing HDR video using 8-bit encoding with additional spatio-temporal dither is impractical because dither patterns do not compress well.

While the dynamic range employed in digital photography is usually limited to 2–3 orders of magnitude, a much broader dynamic range of 4–5 orders of magnitude can be achieved with analog film. The problem of digital encoding, which emulates the dynamic range and *S*-shaped response curve of film has been addressed in patent literature [Lucian et al. 2005]. Custom wavelet encoders, such as layered wavelet encoders, have been designed especially for the purpose of storing wide dynamic range scans of film negatives used in cinematography [Bogart et al. 2003; Demos 2004]. These compression methods however require substantial bit-rates and are not suitable for on-DVD storage or real-time playback. The dynamic range level achieved with analog film and its digital emulation is too low for our purposes. Besides, we argue that the video encoding format should be designed for the capabilities of the human eye rather than analog film or camera characteristics.

Li et al. [2005] propose a wavelet-based tone mapping operator paired with an inverse operator that can restore an original HDR image. Since the tone mapping operator is designed to preserve high frequencies, the restored HDR image is visually very close to the original. The emphasis on high frequencies makes the proposed solution less suitable for JPEG and MPEG encodings, which employ quantization matrices that are perceptually tuned to discard a great deal of visually unimportant high frequencies. Li et al. report relatively poor compression rates when they attempt to use the JPEG encoding of their tone-mapped images. Moreover, a predetermined choice of tone mapping operator is not acceptable for our application and a high computation cost of the wavelet transformation makes this algorithm unsuitable for video (the authors recommend oversampling, i.e., handling the subband computation in the full image resolution to avoid aliasing).

The recent video compression standards offer an extended bit-depth of up to 12 bits for ISO/IEC 14496-2 and ISO/IEC 14496-10 AVC/H.264 with high profiles defined in the Fidelity Range Extensions (FRExt). This unfortunately does not imply that these extensions were designed to store higher dynamic range. Despite the higher bit-depth, the specified transfer functions allow encoding only up to 2.5 log-10 units of dynamic range. Also, encoding at more than 8 bits is rarely implemented. HDR images and video can be encoded with the precision of the human eye using traditional video and image formats that allow the encoding of luminance (transfer function) with eleven or more bits [Mantiuk et al. 2006b]. This has been shown feasible for JPEG2000 [Xu et al. 2005] and MPEG-4 ISO/IEC 14496-2 [Mantiuk et al. 2004]. Even though these approaches offer a straightforward extension to the existing formats, they are not backward compatible and require more than 8-bit decoders, which are expensive and rarely implemented in hardware.

Spaulding et al. [2003] showed that the dynamic range and color gamut of typical sRGB images can be extended using residual images. Their method is backward compatible with the JPEG standard, but only considers images of moderate dynamic range. Ward and Simmons [2004] have proposed a backward-compatible extension of JPEG which enables compression of images of much higher dynamic range (JPEG HDR). An HDR image is tone-mapped and compressed using the standard 8-bit JPEG encoder. An additional *subband* image stores the ratios between the HDR and LDR luminance values that are used to reconstruct the original HDR image. The subband image is downsampled to reduce its size and JPEG compressed. Two outstanding contributions of this work are the maintenance of backward compatibility with standard JPEG and flexibility in choosing an arbitrary tone mapping operator. The proposed HDR MPEG compression achieves both of these goals, and it additionally offers more flexibility in the choice of tone/gamut mapping, better compression and is especially designed for video. We give a detailed comparison between our approach and JPEG HDR in Section 6.

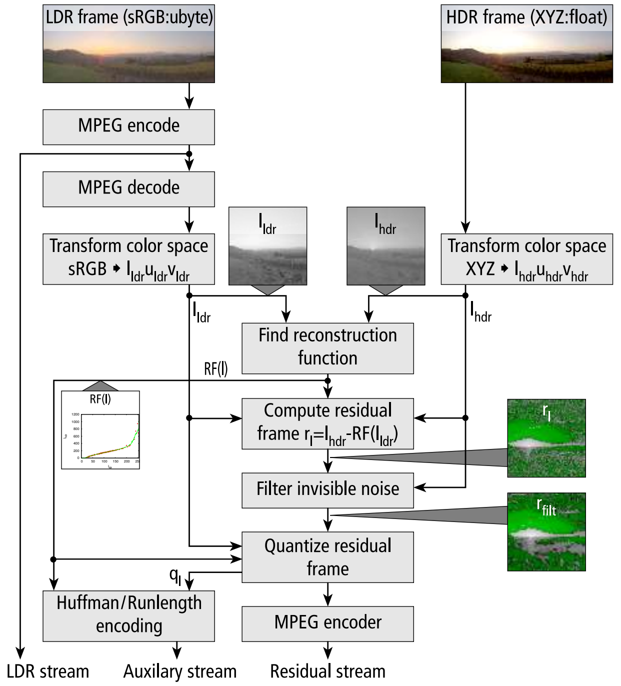

Figure 2: A data flow of backward-compatible HDR MPEG encoding. See text for details.

3 HDR Video Compression

The complete data flow of the proposed backward-compatible HDR video compression algorithm is shown in Figure 2. The encoder takes two sequences of HDR and LDR frames as input. The LDR frames, intended for LDR devices, usually contain a tone-mapped or gamut mapped version of the original HDR sequence. The LDR frames are compressed using a standard MPEG encoder (**MPEG encode** in Figure 2) to produce a backward-compatible LDR stream. The LDR frames are then decoded to obtain a distorted (due to lossy compression) LDR sequence, which is later used as a reference for the HDR frames (see **MPEG decode** in Figure 2). Both the LDR and HDR frames are then converted to compatible color spaces, which minimize differences between LDR and HDR colors. The reconstruction function (see **Find reconstruction function** in Figure 2) reduces the correlation between LDR and HDR pixels by giving the best prediction of HDR pixels based on the values of LDR pixels. The residual frame is introduced to store a difference between the original HDR values and the values predicted by the reconstruction function. To improve compression, invisible luminance and chrominance variations are removed from the residual frame (see **Filter invisible noise** in Figure 2). Finally, the pixel values of a residual frame are quantized (see **Quantize residual frame** in Figure 2) and compressed using a standard MPEG encoder into a residual stream. Both the reconstruction function and the quantization factors are compressed using a lossless arithmetic encoding and stored in an auxiliary stream. The most important steps of the compression algorithm are described in detail in the following subsections while the details, which are sufficient to reimplement the algorithm, are given in the technical report [Mantiuk et al. 2006a].

3.1 Color Space Transformations

Both LDR and HDR frames must be transformed to *compatible* and *perceptually uniform* color spaces to enable any comparison between LDR and HDR pixel values and to assess their correlation. The "compatible" color spaces mean here that color channels of both LDR and HDR pixels represent approximately the same information. Perceptual uniformity is needed to estimate color differences according to perceivable, rather than arithmetic, differences. Furthermore, an HDR color space must represent the full color gamut visible to the human eye. To achieve all these goals, we have derived two color spaces: (i) A color space for LDR pixels that encodes chroma using CIE 1976 Uniform Chromacity Scales (u', v', similar to *logLuv* encoding [Ward Larson 1998]) and luma using sRGB nonlinearity, which consist of a linear and power function segments; (ii) A color space for the HDR pixels uses the same u', v' encoding for chroma as the color space for LDR pixels, and a perceptually uniform luminance encoding. The sRGB nonlinearity cannot be used for luminance values ranging from 10^{-5} to 10^{10} cd/m^2, which can be found in real world scenes. Therefore we apply the luminance encoding that has been derived from the contrast detection measurements for the full visible range of luminance. This encoding was shown to have similar properties to gamma correction for LDR, but can encode luminance values found in HDR images using 11–12 bits and ensures that the quantization error is below the threshold of visibility. A similar encoding was used in the context of HDR video encoding [Mantiuk et al. 2004]. The details and the derivation of this space can be found in [Mantiuk et al. 2006b]. To convert HDR luminance, y, into 12-bit HDR luma, l_{hdr}, we use the formula:

$$l_{hdr}(y) = \begin{cases} a \cdot y & \text{if } y < y_l \\ b \cdot y^c + d & \text{if } y_l \le y < y_h \\ e \cdot \log(y) + f & \text{if } y \ge y_h \end{cases} \tag{1}$$

and for the inverse conversion, from 12-bit luma to luminance, we apply:

$$y(l_{hdr}) = \begin{cases} a' \cdot l_{hdr} & \text{if } l_{hdr} < l_l \\ b'(l_{hdr} + d')^{c'} & \text{if } l_l \le l_{hdr} < l_h \\ e' \cdot \exp(f' \cdot l_{hdr}) & \text{if } l_{hdr} \ge l_h \end{cases} \tag{2}$$

The constants are given in the table below:

$a = 17.554$	$e = 209.16$	$a' = 0.056968$	$e' = 32.994$
$b = 826.81$	$f = -731.28$	$b' = 7.3014e-30$	$f' = 0.0047811$
$c = 0.10013$	$y_l = 5.6046$	$c' = 9.9872$	$l_l = 98.381$
$d = -884.17$	$y_h = 10469$	$d' = 884.17$	$l_h = 1204.7$

Note that we use u' and v' chromaticities rather than u^* and v^* of the $L^*u^*v^*$ color space. Although u^* and v^* give better perceptual uniformity and predict the loss of color sensitivity at low light, they are strongly correlated with luminance. Such correlation is highly undesired in image or video compression. Besides, u^* and v^* could reach high values for high luminance, which would be difficult to encode using an 8-bit MPEG encoder.

3.2 Reconstruction Function

Both LDR and HDR frames contain similar information and are therefore strongly correlated. This is illustrated in Figure 3, which shows how the luma values of an LDR frame relate to the luma values of an HDR frame. The relation is different for each tone mapping algorithm, but in general it follows an approximately linear function with more variance at high values. Uncorrelated pixels at the right end of the l_{ldr} axis are the result of luminance clamping that is applied in many tone mapping algorithms. Local tone mapping usually results in higher variance and therefore a more "noisy" shape of this relation, while global tone mapping results in a direct one-to-one relationship unless some pixel values are clamped.

The goal of most compression methods is to decorrelate data, so that the same information is not encoded twice. To decorrelate LDR

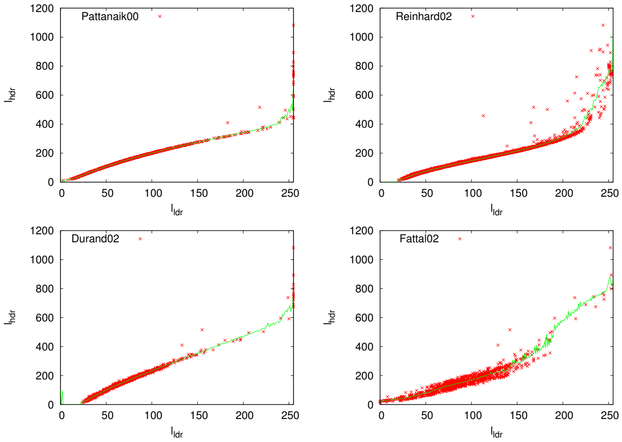

Figure 3: The relation between LDR (l_{ldr}) and HDR (l_{hdr}) luma values for various tone mapping algorithms (marked in red) and the corresponding reconstruction functions (marked in green). Tone mapping algorithms (left to right, top to bottom): [Pattanaik et al. 2000], [Reinhard et al. 2002], [Durand and Dorsey 2002] and [Fattal et al. 2002]. The relations are plotted for the *Memorial Church* image.

and HDR frames, we find a *reconstruction function*, which predicts the value of an HDR pixel based on the value of the corresponding LDR pixel. Having such a function we need only to encode the differences between values predicted by the reconstruction function and the actual values from an HDR frame. Such differences are usually close to zero and therefore can be efficiently encoded in *residual frames*. The reconstruction function needs to be defined for only 256 values (bins) for 8-bit per channel LDR encoding. The function does not need to be continuous since its major role is to make the values of the residual frame as small as possible. Some examples of reconstruction functions for different tone mapping algorithms are plotted in Figure 3 as continuous green lines.

A mapping from LDR values to HDR values is, in the general case, a one-to-many relationship – there are many HDR pixels values that fall in one of 256 bins of the reconstruction function (LDR pixel values). The questions is how to find a value for each bin that would lead to the best compression performance. We experimented with an arithmetic mean, a median and a midrange [1]. While the midrange gave the worst compression ratio, the arithmetic mean and the median exhibited similar performance. We have decided to use an arithmetic mean because of its lower computational cost.

To summarize, we define the reconstruction function as the arithmetic mean of all pixels falling in a corresponding bin Ω_l:

$$RF(l) = \frac{1}{Card(\Omega_l)} \sum_{i\in\Omega_l} l_{hdr}(i) \text{ where } \Omega_l = \{i = 1..N:\ l_{ldr}(i) = l\} \tag{3}$$

$l = 0..255$ is an index of a bin, N is the number of pixels in a frame, $l_{ldr}(i)$ and $l_{hdr}(i)$ are luma values of the i-th LDR and HDR pixel respectively.

We executed a set of tests on video sequences to decide how often a reconstruction function should be updated: each frame, only at each intra-encoded frame (I-frame), or if the update should depend on a difference between consecutive frames. We achieved the best compression ratio when the reconstruction function was updated each frame, while updating it for every I-frame resulted in severe artifacts.

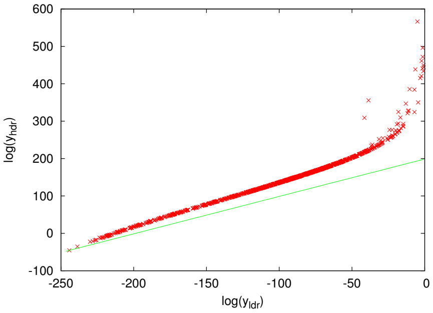

Figure 4: A potential reconstruction function for the approach employed in JPEG HDR [Ward and Simmons 2004] compression (marked in green) and a relation between LDR and HDR pixel values (marked in red). The ratio used in JPEG HDR is equivalent to a linear reconstruction function in the logarithmic domain. Such a function does not decorrelate HDR and LDR luma well and therefore reduces compression savings.

The relation between LDR and HDR frames is complex only for luminance, and color channels can be quite accurately predicted with simple relations $u_{hdr}(i) = u_{ldr}(i)$ and $v_{hdr}(i) = v_{ldr}(i)$. Although this may not be true for some sophisticated gamut mapping cases, we did not find it necessary to compute a reconstruction function for chroma channels for any of the tone mapping operators we tested.

Since the reconstruction function tends to be slowly changing with an increasing slope, we apply an adaptive Huffman algorithm on the differences between the values in consecutive bins to significantly reduce the size of the stored data. The size of the auxiliary data stream, which stores a reconstruction function, is below 1% of the total stream size, therefore its storage overhead is almost insignificant.

We briefly compare our approach with the JPEG HDR compression [Ward and Simmons 2004]. A more detailed comparison will be given in Section 6. The JPEG HDR compression encodes a ratio between HDR and LDR luminance values, rather than a difference between HDR values and the reconstruction function. However, it can be easily shown that such a ratio is meant to achieve the same goal, which is to decorrelate HDR and LDR pixels. Since a ratio of HDR and LDR luminance corresponds to a difference in the logarithmic domain, and our luminance to luma mapping from Equation 1 has roughly logarithmic properties, the ratio encoding of JPEG HDR corresponds to a linear reconstruction function $l_{hdr} = a \cdot l_{ldr}$. As we experimented with such simple reconstruction functions, we found that they give inferior results compared to better fitted ones, like those computed from Equation 3. In Figure 4 we plot the reconstruction function used in JPEG HDR. Obviously, it does not follow the data well and some luma information is therefore encoded twice in an LDR and an HDR (subband) stream, which leads to worse compression performance.

3.3 Residual Frame Quantization

Although the magnitudes of the differences encoded in residual frames are usually small, they can in fact take values from -4095 to 4095 (for 12-bit HDR luma encoding). Such values cannot be encoded using an 8-bit MPEG encoder. Although MPEG standards provide an extension for encoding luma values on 12 bits, such an extension is rarely implemented, especially in hardware. Instead,

[1] Midrange is defined as an arithmetic mean of the maximum and minimum value in a set.

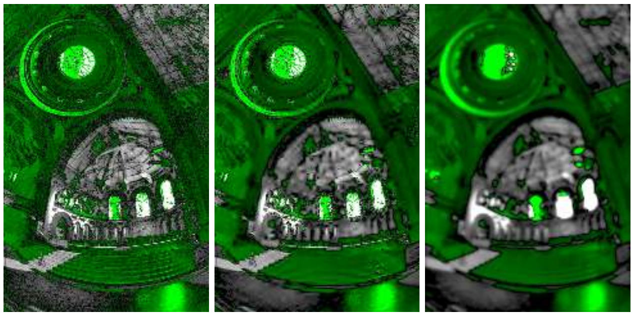

Figure 5: Residual frame before (left) and after (center) filtering invisible noise. Details, such as window frame, are lost when low-pass filtering (or downsampling) is used (right). Green color denotes negative values. The *Memorial Church* image courtesy of Paul Debevec.

we would like to reduce the magnitude of residual values so that they can be encoded using a standard 8-bit MPEG encoder.

We have experimented with a non-linear quantization, where large absolute values of residuals were heavily quantized, while small values were preserved with maximum accuracy. Since few pixels contain a large magnitude of residual, most pixels are not affected by the strong quantization. Such a solution, although giving the best SNR, resulted in poor visual quality for some images. This was because the very few pixels that were heavily quantized attracted attention due to large quantization errors. Therefore the final judgement of quality was mostly based on those few distorted pixels.

A simple clamping of residual values to 8-bit range produced visually better results, but at the cost of losing some details in bright or dark regions. Additionally, to reduce clamping at the cost of a stronger quantization, the residual values can be divided by a constant quantization factor. Such a factor would decide on the trade-off between errors due to clamping and errors due to quantization. Furthermore, we observed that very few bins of a reconstruction function contain residual values that exceed 8-bit range. Therefore the quantization factor can be set separately for each bin, based on the maximum magnitude of the residual that belongs to that bin. Therefore, the residual values after quantization can be computed as:

$$\hat{r}_l(i) = [r_l(i)/q(m)]^{-127\div127}, \text{ where } m = k \Leftrightarrow i \subset \Omega_k \tag{4}$$

and quantization factor, $q(m)$, is selected separately for each bin Ω_k:

$$q(m) = \max(q_{min}, \frac{\max_{i\in\Omega_l}(|r_l(i)|)}{127}) \tag{5}$$

q_{min} is a minimum quantization factor, which is usually set to 1 or 2. $[\cdot]^{-127\div127}$ is an operator that rounds the values to the closest integer and then clamps them if they are smaller than -127 or larger than 127. The l subscript in r_l denotes a luma channel.

The quantization factors $q(m)$, where $m = 0..255$, need to be stored in an MPEG stream to later restore non-quantized residual values on the decoding stage. We store quantization factors together with the reconstruction function in the auxiliary data stream. Since quantization factors are usually equal to q_{min} except for a few bins, we found that a run-length encoding followed by the Huffman encoding can effectively reduce the size of this data.

3.4 Filtering of Invisible Noise

Residual frames do not compress well mainly because they contain a large amount of high frequencies. These high frequencies come from three sources: noise in the source HDR images, rounding errors from the tone mapping algorithm, and the DCT quantization errors due to MPEG encoding of LDR frames (refer to Figure 2). However, much of this high frequency information does not need to be preserved in the residual stream since it is not visible to the human eye. To remove such invisible noise and thus improve compression efficiency, we introduce a filtering algorithm based on a simplified model of the human visual system (HVS). Although models of the HVS have been used before in CG to control rendering [Ferwerda et al. 1997; Bolin and Meyer 1998; Ramasubramanian et al. 1999], the proposed filtering algorithm has been specially designed to handle HDR data and it has been optimized for speed, so that it can efficiently process video sequences. It is also different from a typical denoising algorithms, e.g. [Bennett and McMillan 2005], since it operates on imperceivable, rather than perceivable noise. It can be used as a standard tool which guarantees that all the visual information that cannot be discerned due to imperfections of the human eye and early vision processing will be filtered out from the image.

The standard MPEG encoding already incorporates many aspects of human vision in order to improve compression efficiency. The gamma corrected color space (or transfer function) accounts for luminance masking (sometimes wrongly named the Weber-Fechner law [Mantiuk et al. 2006b]). The limited spatial contrast sensitivity of the HVS is utilized by the DCT quantization matrix. Two different quantization matrices are used for inter- and intra-frames to take advantage of lower sensitivity to high temporal frequencies. However, contrast masking (or visual masking) is very poorly predicted by the mechanism of MPEG encoding. Since contrast masking is primarily responsible for masking invisible high frequency noise, we focus on modeling this aspect of the HVS to filter residual frames.

There are several methods that incorporate visual masking in image encoding algorithms, such as optimized DCT quantization matrices [Ahumada and Peterson 1993; Watson et al. 1994], the *pre-quantization* scheme [Safranek 1993], or the point-wise extended masking in the JPEG-2000 standard [Wenjun et al. 2000]. However, since all these approaches are either not suitable for video or require significant changes in the MPEG encoder/decoder, we decided to use yet another approach, which involves the *prefiltering* of residual frames before they are passed to the MPEG encoder. Prefiltering methods have been shown to improve video compression [Border and Guillotel 2000]. They do not depend on a compression algorithm and therefore do not require any changes to the encoder. The proposed prefiltering algorithm precisely models contrast masking in the wavelet domain, which is quite difficult and inaccurate in the DCT domain. The prefiltering is especially well suited for the residual frames, since they contain mostly low magnitude contrast, while prefiltering involves thresholding of wavelet coefficients that are below the predicted visibility level. If the wavelet coefficients are low, most of them are set to zero and therefore compression efficiency is improved. The prefiltering affects only encoding speed while decoding speed is usually improved due to the reduced stream size.

The input to our residual filtering algorithm consists of two frames: a residual frame (Figure 5 left) and an original HDR frame, which is a masker for the residual. Both frames should be stored in the perceptually uniform luma / chroma color space. Output of the filtering is a residual frame with high frequencies attenuated in those regions where they are not visible (Figure 5 center). The data flow of the algorithm is shown in Figure 6. Though we describe processing that

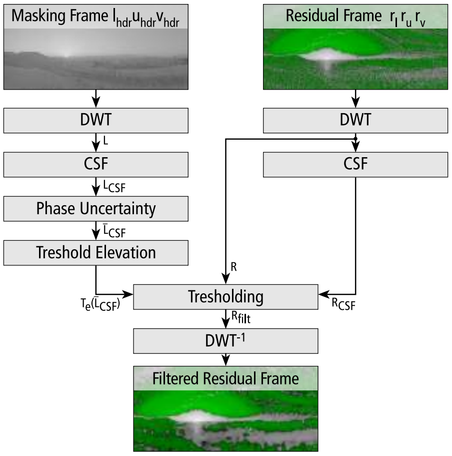

Figure 6: A data flow of the residual frame filtering, which removes imperceptible noise for better compression performance.

is done on a luma channel, the same processing is performed for two chroma channels, which are subsampled to half of their original resolution. This approximately accounts for the differences between the Contrast Sensitivity Function (CSF) for luminance and chrominance.

In the first step we apply the Discrete Wavelet Transform to split a residual frame into several frequency and orientation selective channels. We have experimented with the cortex decomposition [Watson 1987] performed in the Fourier domain, which can better approximate visual channels, but we rejected this approach due to prohibitively long execution times (up to 1 minute per frame). Wavelets, on the other hand, lead to computationally more efficient algorithms and were shown to be useful for modeling many aspects of the HVS [Bradley 1999; Wenjun et al. 2000]. We employ CDF 9/7 discrete wavelet basis which is also used for the lossy compression of JPEG-2000. This wavelet basis provides a good trade-off between smoothness and computational efficiency. We use only the three finest scales of the wavelet decomposition since filtering of lower spatial frequencies at coarser scales could lead to noticeable artifacts.

In the next step we account for lower sensitivity of the HVS for high frequencies, which is usually modelled with the Contrast Sensitivity Function (denoted as **CSF** in Figure 6). We weight each band of wavelet coefficients by a constant value in the same way as is done in JPEG-2000. The weighting factors for a viewing distance of 1,700 pixels (≈ 1.5 × screen height) are given in the table below.

Scale	LH	HL	HH
1	0.275783	0.275783	0.090078
2	0.837755	0.837755	0.701837
3	0.999994	0.999994	0.999988

The visual channels have limited phase sensitivity, ranging from 45° to more than 90°. Because of this, the masking signal affects not only regions where the values of wavelet coefficients are the highest, but may also affect neighboring regions. Phase uncertainty reduces the effect of masking at edges, as opposed to textures which show a high amount of masking. Following the point-wise extended masking in JPEG-2000 [Wenjun et al. 2000], we model phase uncertainty with the $L_{0.2}$-norm:

$$\bar{L}_{CSF} = \frac{1}{Card(\Theta)} \left(\sum_{\Theta} |L_{CSF}|^{0.2} \right)^{\frac{1}{0.2}} \tag{6}$$

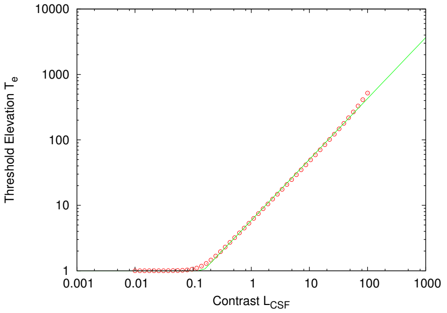

Figure 7: The threshold elevation function for contrast represented as wavelet coefficients. Data points were found from the model proposed by Daly [1993] after conversion to new units of contrast. The solid line is a model from Equation 7, which approximates these data points.

where Θ denotes a neighborhood of a wavelet coefficient (we use a box 13 × 13 kernel in our implementation).

In the following step we predict how contrast thresholds change in the presence of a masking signal, which is an original HDR frame in our case. To model contrast masking, we employ a threshold elevation function, which we derive from the model proposed by Daly [1993] (also used in [Ramasubramanian et al. 1999]). We assume a masking slope of 1.0, which was shown to be appropriate for natural images [Mantiuk et al. 2005]. We modify the original threshold elevation function to make it applicable to the perceptually uniform luma space, which we introduced in Section 3.1. Threshold elevation for this space can be approximated by the function:

$$T_e(\bar{L}_{CSF}) = \begin{cases} 1 & \text{if } \bar{L}_{CSF} \leq a \\ c \cdot (\bar{L}_{CSF}))^b & \text{otherwise} \end{cases} \tag{7}$$

where $\bar{L}_{CSF}$ is a wavelet coefficient, $a = 0.093071$, $b = 1.0299$ and $c = 11.535$. The function with original data points is plotted in Figure 7.

Next, we compare each CSF weighted coefficient of a residual frame, R_{CSF}, with the corresponding value of the threshold elevation T_e. If the residual is smaller than the visibility threshold predicted by the threshold elevation function from Equation 7, we can safely set this coefficient to zero without introducing visually noticeable changes. Formally, it can be written as:

$$R_{filt} = \begin{cases} 0 & \text{if } T_e(\bar{L}_{CSF}) < R_{CSF} \\ R & \text{otherwise} \end{cases} \tag{8}$$

Finally, we transform the filtered wavelets coefficients, R_{filt} back to the image domain (**DWT**$^{-1}$ in Figure 6).

The prefiltering method presented above can substantially reduce the size of a residual stream and is a reasonable trade-off between computational efficiency and accuracy of the visual model. The encoding time is affected by no more than 80% when filtering is used and it can only reduce decoding times because of a smaller resulting bit-stream. We have resolved to simplify some aspects of the visual model in order to bring the performance to an acceptable level. For example, we do not model the Optical Transfer Function (OTF) since we found that its local effect is negligible (close or below the MTF of a monitor) for typical viewing conditions and the flare effect would require much larger kernels or operations in the Fourier domain, which would slow down filtering significantly. For performance reasons we use wavelets, which do not model visual channels as accurately as other transformations designed especially for that purpose. Since we do not have precise information on the optical flow, we can not model temporal aspects of the CSF. The temporal CSF is partially taken into account by the MPEG encoder. Nevertheless, the proposed prefiltering method takes into account more perceptual factors than most state-of-the-art video compression techniques and can additionally handle HDR scenes.

Note that this is not the only possible filtering scheme and some applications may use different filters. For example, sub-sampling and reducing the resolution of residual frames, as done in [Ward and Simmons 2004], can improve both compression efficiency and encoding/decoding speed, but at the cost of blurry artifacts, especially in the regions where LDR pixels have been clamped to minimum or maximum values. If video is to be displayed on a particular type of display, there is no reason to encode the information that can not be displayed. Therefore the filter can take into account the limitations of the display, which are usually more restrictive than the full capabilities of the HVS.

4 Implementation Details

The implementation of MPEG-4 Advanced Simple Profile ISO/IEC 14496-2, available from `http://www.xvid.org/`, was used as a base MPEG encoder/decoder. However, our method is not restricted to any particular implementation and any other video or image encoder could be used instead. The backward-compatible HDR encoder/decoder has been implemented as a dynamic library to simplify integration with external software. We separately implemented a set of command-line tools for encoding and decoding video streams to and from HDR image files and integrated them with the *pfstools* framework (`http://pfstools.sourceforge.net/`). An LDR stream can be played back using any video player capable of decoding MPEG-4 video. To play back an HDR stream, we have developed a custom HDR video player, which can display video on both LDR and HDR displays [Seetzen et al. 2004].

Since HDR video playback involves decoding two MPEG-4 streams, an LDR and a residual stream, achieving an acceptable frame rate is more challenging than in the case of an ordinary LDR video. To boost playback frame rate, we moved some parts of the decoding process to graphics hardware. We found that both color space conversion and up-sampling of color channels are computationally expensive when executed on a CPU while the same operations can be performed in almost no time on a GPU as fragment programs. The remaining parts of the decoding and encoding algorithm were implemented using the *SSE* instruction set whenever possible. Additionally, some color conversion functions were significantly accelerated with the use of fixed point arithmetic and lookup tables. All those optimizations let us achieve real-time software playback of HDR movies (25–50 frames per second for the VGA resolution, depending on a hardware configuration and quality settings of the compression).

5 Results

To test the performance of our backward-compatible HDR MPEG compression, we have executed an extensive set of over 1,500 tests on images and video sequences. A good video compression should produce a video stream of the smallest size (measured in our tests as the number of bits per pixel) at the highest quality. Although simple arithmetic metrics, such as Signal to Noise Ratio (SNR), are usually used to measure the quality of compressed images, we follow a common practice in CG [Ward and Simmons 2004; Xu et al. 2005] and also use advanced metrics that account for the aspects of the HVS. We used the following metrics to evaluate the quality of the decoded images and video sequences:

HDR VDP — Visual Difference Predictor for High Dynamic Range images [Mantiuk et al. 2005]. This is a fidelity metric that can predict the differences between two images that are likely to be noticed by a human observer. This metric has been especially designed for HDR images and takes into account such effects as light scattering in the optics of the eye, luminance masking for the visible range of luminance, spatial contrast sensitivity, local adaptation and visual masking. The result of the HDR VDP is a probability of detection map, which assigns for each pixel a probability that the difference can be noticed. For easier interpretation of the results we have summarized the prediction of the HDR VDP with a single number, which is a percentage of pixels in an image that exceed 75% probability of detection. The lower percentage denotes a better quality, as fewer pixels are noticeably affected by compression distortions. We used the original implementation of the HDR VDP provided by the authors.

UQI — Universal Image Quality Index [Wang and Bovik 2002]. This quality metric models any image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. The index, although it does not employ any model of the HVS, shows consistency with a subjective quality measurement and performs better than the mean squared error. The quality index can range from -1 (the worst quality) to 1 (the best quality). We have implemented this metric according to the original paper [Wang and Bovik 2002]. To adapt this metric to HDR images, we provide for input luma values computed with Equation 1.

SNR — Signal to Noise Ratio. This is the simplest but also the most commonly used metric, which does not model any aspects of the HVS and may not be consistent with a subjective quality measurement. We used the standard formulas to compute the SNR for the luma values computed with Equation 1. The larger value of SNR usually results in higher quality.

In the following sections we analyze several aspects of our encoding scheme based on the collected test results.

5.1 Influence of Tone Mapping Operator

Although there are no restrictions on tone mapping / gamut mapping or stylizing used to obtain LDR frames, the choice of such processing will obviously affect the efficiency of compression. We tested our encoder with five tone mapping operators (TMOs) from the *pfstmo* package[2] (labels in italics): *Pattanaik00* — Time-Dependent Visual Adaptation [Pattanaik et al. 2000]; *Durand02* — Fast Bilateral Filtering [Durand and Dorsey 2002]; *Reinhard02* — Photographic Tone Reproduction [Reinhard et al. 2002]; *Fattal02* — Gradient Domain [Fattal et al. 2002]; *Drago03* — Adaptive Logarithmic Mapping [Drago et al. 2003]. We used the default parameters for all TMOs. To prevent temporal flickering in tone-mapped video sequences, we added extensions to the original TMOs that ensured time-coherence of the TMO parameters. The extension ensured that the maximum difference of selected parameters (e.g. L_{White} for the *Reinhard02* TMO) between frames is always below the visibility threshold.

Figure 8 shows how the efficiency of compression is affected by a TMO. The results for most TMOs are in fact similar, with the exception of *Fattal02*, which results in significantly larger streams. This is mainly because the operator introduces the largest changes of local contrast in LDR frames, which results in the high variance of residual values. The result is consistent with our earlier considerations in Section 3.2, which suggested that global TMOs are better approximated by the reconstruction function and therefore result in

[2] More details on the *pfstmo* at: `http://www.mpii.mpg.de/resources/tmo/`

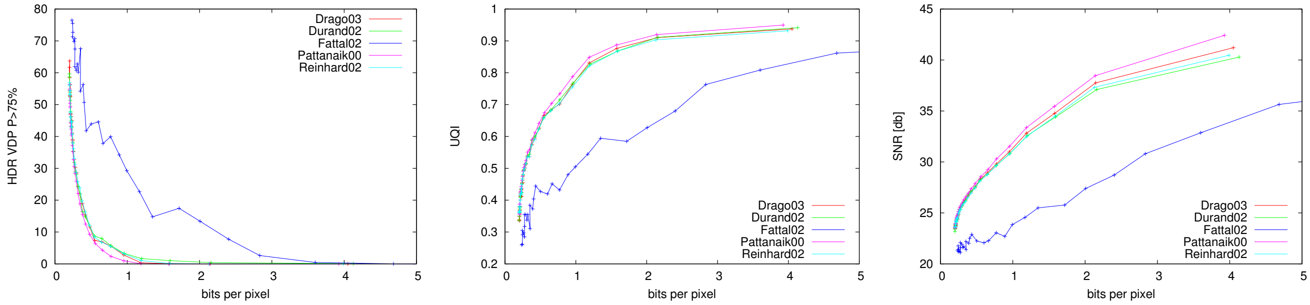

Figure 8: Comparison of compression performance for different tone mapping operators. See Section 5 for the description of the quality metrics. "+" denotes measurement points for a selected image.

smaller magnitudes of the residual. If *Fattal02* is used to generate LDR video, the size of the LDR stream is also affected since high frequencies, which are poorly compressed by the MPEG encoding, are enhanced (we expect similar problems with the tone mapping approach proposed by Li et al. [2005]). Nevertheless, *Fattal02* gave the most attractive LDR images. Therefore the selection of a proper TMO for compression is often a combined aesthetic and economic choice.

5.2 The Effect of Invisible Noise Filtering

We validate the algorithm for filtering invisible noise, described in Section 3.4, for a range of MPEG quality settings. Figure 9 illustrates how the size of a residual stream is reduced when the filtering is used. Note that the largest savings are possible for the best quality settings. This is because the strength of the filtering is determined by the visibility thresholds, which do not depend on quality settings. The filtering has a minimal impact on the stream size for low quality settings since the distortions introduced by the aggressive DCT quantization are far above the visibility thresholds used in the filtering. Figure 13 shows how both the total stream size and quality are affected when the residual frames are filtered. Although the filtering in fact introduces changes that are detected by the HDR VDP (probably due to a mismatch in the visual models used by the filtering and the HDR VDP), the loss of quality is fully compensated by the bit-rate savings (see Figure 13). Moreover, we observe that the subjective quality of filtered video is better than predicted by the HDR VDP. This is because the blurry artifacts due to the wavelet based filtering are less objectionable than blocky artifacts of DCT coding (see Figure 12). Although HDR VDP can predict the existence of visible distortions, it can neither estimate their magnitude, nor their impact on perceived quality.

5.3 Comparison with Lossy HDR Compression Methods

The performance of the proposed method (labeled as **HDR MPEG**) has been compared with two others lossy HDR compression methods:

HDRV — Perception-motivated HDR Video Encoding [Mantiuk et al. 2004]. This is the first lossy HDR video compression method, which, however, does not offer backward compatibility. The method encodes HDR pixels using 11 bits for luminance and twice 8 bits for chrominance. Since the resulting video stream does not contain any information on LDR frames, it can be expected that this compression method gives better results than backward-compatible methods. We used the original implementation provided by the authors.

JPEG HDR — Subband encoding of high dynamic range imagery [Ward and Simmons 2004; Ward and Simmons 2005]. This is a backward-compatible HDR image encoding, which is conceptually the closest to our method. A detailed comparison of both our approach and JPEG HDR is given in Section 6. The method involves sub-sampling of a subband layer (a residual frame in the case of our encoding), which can lead to the loss of high frequencies. To prevent this loss, the method suggests three approaches: **pre-correction** of LDR layer, to encode within this layer high frequencies that can be lost due to sub-sampling; **post-correction** which tries to restore high frequencies that have been lost rather than modifying LDR image, and **full-sampling**, which means that no sub-sampling is performed. We used the original encoding/decoding library provided by the authors.

To evaluate the performance of intra-frame (image) compression, we ran the tests on eight representative HDR images. We chose the *Reinhard02* TMO to compare our algorithm with other lossy compression methods. This TMO performed similar to the others and is also used in JPEG HDR. Figure 14 shows the averaged results. The HDRV encoding clearly shows the best performance for all three quality metrics. This can be explained by the lack of any information on an LDR stream, which reduces the amount of information that needs to be stored but also makes this encoding incompatible with the LDR MPEG format. For the HDR VDP and the UQI, JPEG HDR performs almost the same as our method for the pre-correction and the post-correction approach, but is worse for the full-sampling. Note that our compression method does not involve sub-sampling and therefore is closer to the full-sampling than the other two approaches. JPEG HDR performs worse than our method for the SNR metric. The improved performance of our encoding over JPEG HDR for images is surprising, since the image encoding algorithms, such as JPEG, are known to perform better than intra-frame video encoding. This is due to better arithmetic encoding and a quantization matrix, which is especially optimized for images. Another difference between two methods that affects the performance is that HDR MPEG encodes information on all color channels in the residual stream while the JPEG HDR encodes only luminance in the additional subband layer (see details in Section 6)

The performance of inter-frame (video) compression was tested on two video sequences for both HDR MPEG and HDRV, while JPEG HDR was not included in these tests. Since both the VDP and the UQI are designed for images and are less suitable for video (large computational cost, lack of temporal aspects), we computed the SNR over all video frames to measure quality. The averaged results for two video sequences are shown in Figure 10. Similarly as for images, HDRV gave better SNR than HDR MPEG for the same number of bits. HDR MPEG, however, could achieve a higher SNR than HDRV for very high bit-rates.

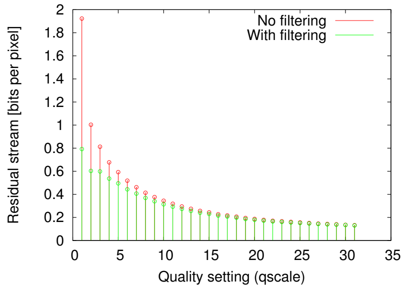

Figure 9: The size of a residual stream with and without invisible noise filtering with respect to the quality settings. The largest savings are achieved for the best quality settings.

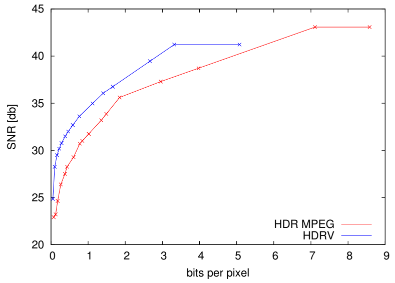

Figure 10: Comparison of lossy HDR compression algorithms. Averaged results for two video sequences.

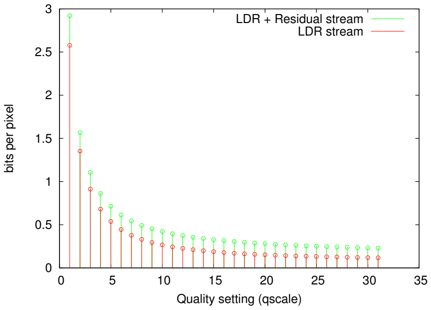

Figure 11: The size of a backward-compatible HDR stream (LDR+Residual+Auxiliary) compared with the size of the LDR stream alone with respect to the quality settings (low *qscale* denotes high quality) for the *Reinhard02* TMO. Results averaged over a set of images.

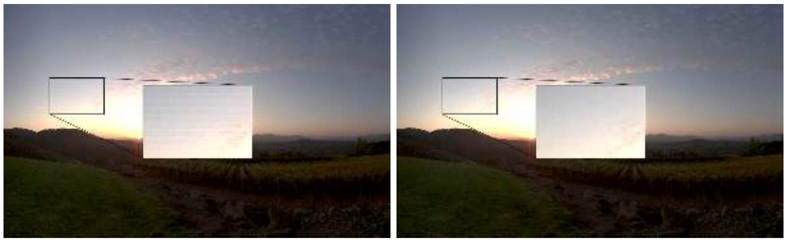

Figure 12: Quality comparison for an image compressed without filtering (left) and with invisible noise filtering (right). Both images were compressed to streams of approximately the same size. The strongly visible blocky artifacts in the image compressed without filtering become barely noticeable in the “filtered” image. Note that the artifacts may not be visible in print and should be observed on a gamma-2.2 monitor. No filtering: $qscale = 6$, bpp $= 1.37$, HDR VDP 75%$= 3.12\%$; With filtering: $qscale = 2$, bpp $= 1.23$, HDR VDP 75%$= 1.11\%$.

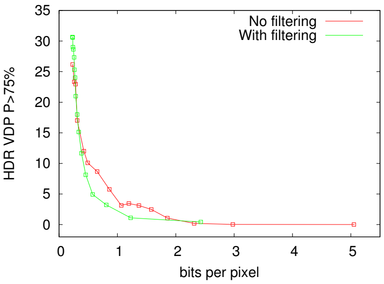

Figure 13: Performance of HDR video compression with and without invisible noise filtering (refer to Section 3.4). The bit-rate savings gained on the filtering compensate for the loss of quality.

5.4 The Cost of Encoding Residual Stream

The proposed HDR encoding method is designed to be an extension to the existing MPEG formats. Therefore, it is interesting to know how much more data must be stored to include additional HDR information. We plot the size of the total HDR stream (LDR + Residual + Auxiliary stream) against the size of the LDR stream in Figure 11. The residual stream does not seem to depend on the quality settings as much as the LDR stream. Therefore its share in the total stream size is the smallest for high quality settings. This can be expected since the residual stream encodes the difference between LDR and HDR frames, including those differences that result from lossy compression of the LDR stream (refer to the MPEG encoding and decoding stages in Figure 2). The lower quality LDR stream means that more information needs to be stored in the residual stream. Overall, the share of residual stream ranges from 5 to 70 percent, depending on the image, quality settings and a TMO. A well chosen TMO and a decent quality settings result in a residual stream that is 25–30% of the LDR stream. The size of the auxiliary stream is negligible.

6 Discussion

Although the proposed backward-compatible HDR encoding algorithm seems to be conceptually similar to the JPEG HDR compression [Ward and Simmons 2004], there are several important differences between the approaches, which not only enable video compression, but also result in better compression and more flexibility of HDR MPEG. As discussed in Section 3.2, HDR MPEG can adapt the reconstruction function to the tone/gamut mapping algorithm used to generate LDR frames and therefore reduces the magnitude of the residual values. This results in better compression ratios as compared to JPEG HDR (refer to Section 5.3), although the results would be even more favorable if we had used the JPEG algorithm instead of MPEG intra-frame compression to encode images. Further bit-rate savings in MPEG HDR come from perceptually optimized color spaces for HDR pixels (refer to Section 3.1).

HDR MPEG offers perceptually conservative and time coherent encoding of residual values, while JPEG HDR suggests an ad-hoc approach to encoding subband, which is not suitable for video. The JPEG HDR encoder transforms subband values to the logarithmic domain and then linearly scales them so that the minimum and the maximum values fit in the 0–255 range. Since the minimum and the maximum subband value can differ from image to image, the scaling factor can also change from frame to frame for video sequences, which would result in temporal flickering and lack of temporal coherence in subband frames. Such a lack of temporal coherence can significantly impact the performance of MPEG inter-frame compression. HDR MPEG, on the other hand, guarantees the temporal coherence of residual frames. Moreover, the linear scaling of subband values in JPEG HDR makes the quantization of the subband layer difficult to predict and control. JPEG HDR will quantize subband values with high accuracy for those images that lead to small magnitude of subband values, perhaps wasting some bit-rate on in-

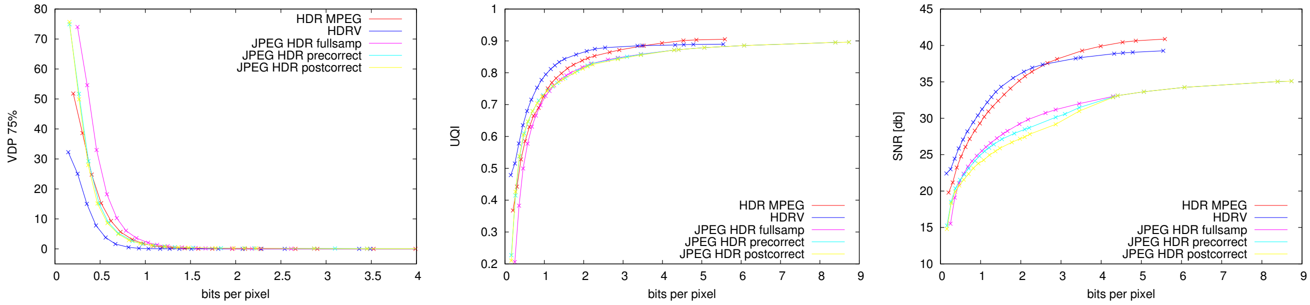

Figure 14: Comparison of lossy HDR compression algorithms. Averaged results for a set of images.

visible contrast details. For another set of images, which result in large magnitude of subband values, JPEG HDR can quantize too coarsely, leading to contouring artifacts. To reduce quantization, JPEG HDR skips a small percent of the brightest and the darkest pixels in an image, which however can lead to loss of some details (see Figure 15). HDR MPEG quantizes color values consistently for consecutive frames and the quantizer is based on the visibility thresholds of the HVS rather than frame content.

Unlike JPEG HDR, the proposed compression method does not impose any restrictions on the choice of a TMO and a gamut mapping algorithm. A TMO for HDR MPEG can saturate both luminance and color, change color values and enhance local contrast. Such changes may result in a lower compression ratio, but both LDR and HDR frames will be preserved in the resulting video stream. JPEG HDR will lose most color differences between HDR and LDR since it does not store color in the subband layer. Such unrestricted control over the appearance of both LDR and HDR streams is very important for our major application - a storage format for digital movies whose appearance cannot be compromised.

Finally, a sub-sampling of the subband layer in JPEG HDR may lead to the loss of visible details. Although the *pre-correction* may be used to avoid loss of high frequency details, this leads to distorted LDR frames, which, similar to the companding approach [Li et al. 2005], is not acceptable for applications requiring uncompromised quality of tone-mapped images. The *post-correction*, although it does not modify the source image, also does not give as good results as the *pre-correction*. *Full-sampling*, on the other hand, does not give as good compression ratio as the other two approaches. The counterpart of sub-sampling in the proposed HDR MPEG is filtering of invisible noise (see Section 3.4). The filtering has a similar goal as the sub-sampling — to reduce high frequency noise and improve compression, but it does it in a more selective manner. The proposed filtering removes only those high frequency details which are not visible and therefore can be smoothed out without impairing the visual quality of the resulting video. Sub-sampling, obviously, cannot give such a guarantee (refer to Figure 5).

7 Conclusions and Future Work

In this work we propose a first backward-compatible HDR MPEG video compression method that can facilitate a smooth transition from LDR to HDR content. The storage cost of a backward-compatible HDR stream is modest (about 30% overhead), compared to the huge storage requirement of High Definition video. The proposed format is especially suitable for DVD movie distribution, which must ensure the compatibility with existing DVD players that are not capable of HDR playback. The format design conforms to standard 8-bit MPEG decoding chips. The method allows for separate tone/gamut mapping of LDR and HDR video, which is essential for top-quality movie production. We introduced a pair of compatible color spaces that facilitate comparisons between LDR and HDR pixels. The nonlinear function used to encode HDR luminance can be regarded as an extended “gamma correction” that can be used for the full range of visible luminance values. To achieve even better compression performance, we employed an advanced model of the HVS, which is tuned for the full range of visible luminance and is suitable for HDR image processing. We introduced an HDR filtering solution based on this model which selectively and conservatively removes imperceptible high-frequency details from the video stream prior to its compression. We believe that our computationally efficient HVS model and HDR filtering solution are general enough to find other applications in computer graphics and digital imaging.

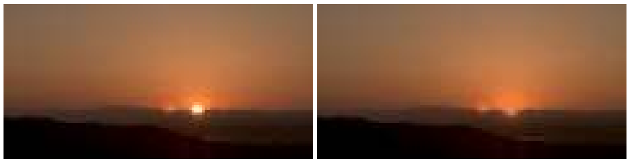

Figure 15: Very bright pixels in the original image (left) are lost after compression with JPEG HDR (right) at quality setting 90 and with the *precorrection* (default settings). This is because a small percentage of the brightest and darkest pixels is skipped when computing minimum and maximum value of the subband image.

We implemented and tested a dual video stream encoding for the purpose of a backward-compatible HDR encoding, however, we believe that other applications that require encoding multiple streams can partly or fully benefit from the proposed method. For example, a movie could contain a separate video stream for color blind people. Such a stream could be efficiently encoded because of its high correlation with the original color stream. Movie producers commonly target different audiences with different color appearance (for example *Kill Bill Vol. 2* was screened with a different color stylization in Japan). The proposed algorithm could be easily extended so that several color stylized movies could be stored on a single DVD. This work is also a step towards an efficient encoding of multiple viewpoint video, required for 3D video [Matusik and Pfister 2004].

8 Acknowledgments

We thank Greg Ward and Helge Seetzen for making us interested in the problem of backward compatibility in HDR encoding and inspiring discussions. We also thank Grzegorz Krawczyk, Kaleigh

Smith, and Peter Longhurst for the helpful comments on the paper, and Jan Petersen and Stoyan Mutafchiev for help in preparing graphics and video material. This research was partly supported by BrightSide Technologies.

References

AHUMADA, A., AND PETERSON, H. 1993. Luminance-model-based DCT quantization for color image compression. In *Human Vision, Visual Processing and Digital Display*, SPIE, volume 3299, 191–201.

BENNETT, E. P., AND MCMILLAN, L. 2005. Video enhancement using per-pixel virtual exposures. *ACM Trans. on Graph. 24*, 3, 845–852.

BOGART, R., KAINZ, F., AND HESS, D. 2003. OpenEXR image file format. In *ACM SIGGRAPH 2003, Sketches & Applications*.

BOLIN, M. R., AND MEYER, G. W. 1998. A perceptually based adaptive sampling algorithm. In *Proc. of ACM SIGGRAPH 1998*, 299–309.

BORDER, P., AND GUILLOTEL, P. 2000. Perceptually adapted MPEG video encoding. In *Proc. of Human Vision and Electronic Imaging V*, SPIE, volume 3959, 168–175.

BRADLEY, A. P. 1999. A wavelet visible difference predictor. *IEEE Transactions on Image Processing 8*, 5, 717–730.

DALY, S. J., AND FENG, X. 2003. Bit-depth extension using spatiotemporal microdither based on models of the equivalent input noise of the visual system. In *Color Imaging VIII: Processing, Hardcopy, and Applications*, SPIE, volume 5008, 455–466.

DALY, S. J., AND FENG, X. 2004. Decontouring: Prevention and removal of false contour artifacts. In *Proc. of Human Vision and Electronic Imaging IX*, SPIE, vol. 5292, 130–149.

DALY, S. 1993. The Visible Differences Predictor: An algorithm for the assessment of image fidelity. In *Digital Image and Human Vision*, Cambridge, MA: MIT Press, A. Watson, Ed., 179–206.

DEMOS, G. 2004. High quality, wide dynamic range, compression system. In *SMPTE Technical Conference Proceedings*.

DRAGO, F., MYSZKOWSKI, K., ANNEN, T., AND CHIBA, N. 2003. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum, Proceedings of Eurographics 2003 22*, 3, 419–426.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. on Graph. 21*, 3, 257–266.

FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient domain high dynamic range compression. *ACM Trans. on Graph. 21*, 3, 249–256.

FERWERDA, J. A., SHIRLEY, P., PATTANAIK, S. N., AND GREENBERG, D. P. 1997. A model of visual masking for computer graphics. In *Proc. of ACM SIGGRAPH 1997*, 143–152.

KRAWCZYK, G., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2005. Perceptual effects in real-time tone mapping. In *SCCG '05: Proc. of the 21st Spring Conference on Computer Graphics*, 195–202.

LI, Y., SHARAN, L., AND ADELSON, E. H. 2005. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. on Graph. 24*, 3, 836–844.

LUCIAN, I., FELICIA, S., CHARLES, S., AND SIEFKIEN, H. 2005. Digital encode and method of encoding high dynamic range video images. In *US Patent 6,867,717*.

MANTIUK, R., KRAWCZYK, G., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2004. Perception-motivated high dynamic range video encoding. *ACM Trans. on Graph. 23*, 3, 730–738.

MANTIUK, R., DALY, S., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2005. Predicting visible differences in high dynamic range images - model and its calibration. In *Proc. of Human Vision and Electronic Imaging X*, SPIE, volume 5666, 204–214.

MANTIUK, R., EFREMOV, A., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2006. Design and evaluation of backward compatible high dynamic range video compression. MPI Technical Report MPI-I-2006-4-001.

MANTIUK, R., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2006. Lossy compression of high dynamic range images and video. In *Proc. of Human Vision and Electronic Imaging XI*, SPIE, San Jose, USA, vol. 6057 of *Proceedings of SPIE*, 60570V.

MATUSIK, W., AND PFISTER, H. 2004. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Trans. on Graph. 23*, 3, 814–824.

PATTANAIK, S., TUMBLIN, J., YEE, H., AND GREENBERG, D. 2000. Time-dependent visual adaptation for realistic image display. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press, New York, NY, USA, Computer Graphics Proceedings, Annual Conference Series, 47–54.

RAMASUBRAMANIAN, M., PATTANAIK, S. N., AND GREENBERG, D. P. 1999. A perceptually based physical error metric for realistic image synthesis. In *Proc. of ACM SIGGRAPH 1999*, 73–82.

REINHARD, E., STARK, M., SHIRLEY, P., AND FERWERDA, J. 2002. Photographic tone reproduction for digital images. *ACM Trans. on Graph. 21*, 3, 267–276.

REINHARD, E., WARD, G., PATTANAIK, S., AND DEBEVEC, P. 2005. *High Dynamic Range Imaging. Data Acquisition, Manipulation, and Display*. Morgan Kaufmann.

SAFRANEK, R. J. 1993. JPEG compliant encoder using perceptually based quantization. In *Human Vision, Visual Processing, and Digital Display IV*, SPIE, volume 1913, 117–126.

SEETZEN, H., HEIDRICH, W., STUERZLINGER, W., WARD, G., WHITEHEAD, L., TRENTACOSTE, M., GHOSH, A., AND VOROZCOVS, A. 2004. High dynamic range display systems. *ACM Trans. on Graph. 23*, 3, 757–765.

SPAULDING, K. E., WOOLFE, G. J., AND JOSHI, R. L. 2003. Using a residual image to extend the color gamut and dynamic range of an sRGB image. In *Proc. of IS&T PICS Conference*, 307–314.

WANG, Z., AND BOVIK, A. 2002. A universal image quality index. *IEEE Signal Processing Letters 9*, 3, 81–84.

WARD, G., AND SIMMONS, M. 2004. Subband encoding of high dynamic range imagery. In *APGV '04: 1st Symposium on Applied Perception in Graphics and Visualization*, 83–90.

WARD, G., AND SIMMONS, M. 2005. JPEG-HDR: A backwards-compatible, high dynamic range extension to JPEG. In *Proceedings of the 13th Color Imaging Conference*, 283–290.

WARD LARSON, G. 1998. LogLuv encoding for full-gamut, high-dynamic range images. *Journal of Graphics Tools 3*, 1, 815–30.

WATSON, A. B., SOLOMON, J. A., AHUMADA, A., AND GALE, A. 1994. DCT basis function visibility: Effects of viewing distances and contrast masking. In *Human Vision, Visual Processing, and Digital Display V*, SPIE, volume 2179, 99–108.

WATSON, A. 1987. The cortex transform: Rapid computation of simulated neural images. *Comp. Vis. Graph. and Image Proc. 39*, 311–327.

WENJUN, Z., DALY, S., AND SHAWMIN, L. 2000. Visual optimization tools in JPEG 2000. In *IEEE Intern. Conf. on Image Processing*, 37–40.

XU, R., PATTANAIK, S., AND HUGHES, C. 2005. High-dynamic range still-image encoding in JPEG 2000. *IEEE Comp. Graph. and Appl. 26*, 6, 57–64.