EUROGRAPHICS 2014 / B. Lévy and J. Kautz
(Guest Editors)

Volume 33 (2014), Number 2

Perceptual Depth Compression for Stereo Applications

Dawid Pająk,[1] Robert Herzog,[2] Radosław Mantiuk,[3] Piotr Didyk,[4] Elmar Eisemann,[5] Karol Myszkowski,[2] and Kari Pulli[1]

[1]NVIDIA, [2]MPI Informatik, [3]West Pomeranian University of Technology, [4]MIT CSAIL, [5]TU Delft

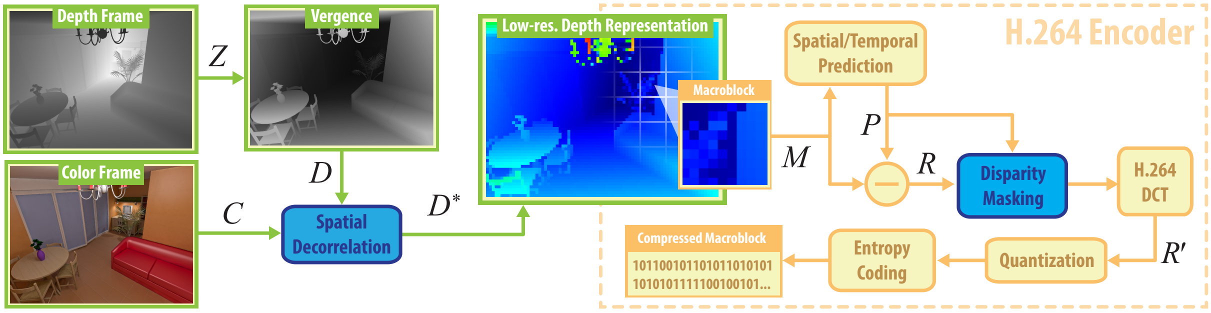

Figure 1: *Outline of our compression framework. H.264 compression tools are framed in orange, our enhancements in blue.*

Abstract

Conventional depth video compression uses video codecs designed for color images. Given the performance of current encoding standards, this solution seems efficient. However, such an approach suffers from many issues stemming from discrepancies between depth and light perception. To exploit the inherent limitations of human depth perception, we propose a novel depth compression method that employs a disparity perception model. In contrast to previous methods, we account for disparity masking, and model a distinct relation between depth perception and contrast in luminance. Our solution is a natural extension to the H.264 codec and can easily be integrated into existing decoders. It significantly improves both the compression efficiency without sacrificing visual quality of depth of rendered content, and the output of depth-reconstruction algorithms or depth cameras.

1. Introduction

Stereo imaging is an important trend in modern media. The amount of available 3D content is increasing and, especially, on a consumer level it has gained much importance. Yet, creating effective 3D content for different displays from movie theaters to hand-held auto-stereoscopic displays is difficult, so that neither viewing comfort nor depth perception is sacrificed. Discomfort typically results from accommodation (focusing distance) and vergence (the distance at which eye viewing directions meet) [LIFH09,SKHB11] disagreements, which is addressed by fitting the scene within a limited comfort zone [LHW*10, KHH*11, MWA*13]. More precisely, the pixel distance between the same scene point in the left and right views is adjusted, hereby influencing vergence.

The required modifications depend on device (type, screen size, and resolution), viewer (left-right eye distance, observer distance), and content [OHB*11] (extent of the scene, visible objects). In a real-time rendering context, the rendering engine can perform such adjustments, but for played-back encoded content (server-side rendering broadcasted to multiple devices, movies, stereo cameras, etc.) such a solution is impossible. Nonetheless, warping operations, based on color images with per-pixel depth (considered by several standards, e.g., HDMI 1.4), can efficiently produce an appropriate high-quality view [DRE*10]. Color + Depth even compresses better than two color images [MWM*09]. Using multiple color images and a depth map, or even multiple depth maps, can increase the warping quality, and is beneficial for multi-view systems [HWD*09]. Such solutions can reduce the server workload, as multiple heterogeneous clients can receive the same information stream, which is locally adapted to the viewing conditions, instead of having the server provide distinct content for each observer. Furthermore, depth information also enables inserting commercials or synthesizing new views from arbitrary positions as required in 3DTV applications, but increase bandwidth compared to a standard video.

Encoding color-converted depth via standard codecs [PKW11] is wasteful. Color and depth have a strong correlation, but it is rarely exploited. Other recent solutions only work for rendered content [PHE*11]. Finally, no monocular video encoding method considers the perceptual quality of stereo, thereby risking compression artifacts, potentially introducing viewing discomfort.

We present three major contributions:

- measurements of neighborhood masking (in terms of spatial location and spatial frequencies);
- a corresponding model, applicable in various contexts;
- a method for simultaneous depth/color compression integrated into existing H.264 encoders.

Our compression exploits the fact that depth edges often coincide with color edges. But if a color edge is missing, the depth difference is often not visible either. Our solution upsamples a special low-resolution depth stream using the color image and can be integrated easily into standard codecs (e.g., H.264) by using our new neighborhood-masking model to reduce residuals in the frame prediction.

2. Previous work

In this section, we discuss existing solutions for depth compression and give an overview of attempts to explore visual masking for compression. More on left/right RGB video stream compression, asymmetric coding, and multi-view compression can be found in an excellent survey [DHH11].

Depth-map compression The H.264/AVC standard enables simultaneous (but independent) depth handling via the "Auxiliary Picture Syntax" [MWM*09], yet encoding is optimized for human perception of natural images. There is no mechanism concentrating on depth quality [MMS*09], which is also true when encoding depth in color channels [PKW11]. Further, bandwidth control is unintuitive; in some sequences depth bitrates have more influence on the quality of a warped image than the bitrate of the color image [MWM*09, Fig. 3], while high depth quantization can be tolerated in others [HWD*09, Fig. 6] and regions-of-interest strategies are even harder to control [KbCTS01].

Multi-view video codecs, such as H.264 MVC, encode views in separate streams and exploit spatio-temporal redundancy between neighboring views. When compared to independent compression of views with H.264 AVC, multi-view coding reduces the bit-rates by $15\% - 25\%$ [DGK*12]. Nonetheless, this still can be a significant cost in terms of bandwidth and computation. For example, autostereoscopic displays require multiple UHD video streams, leading to prohibitively expensive transmission and coding times with H.264 MVC [VTMC11]. Further, image edges are ignored, which can lead to problems for high-quality 3D videos. Our solution can be integrated in such frameworks and would improve compression ratios and fidelity.

Similarly, previous work rarely considers particularities of depth information. Depth is usually a smooth signal with abrupt discontinuities at object boundaries [WYT*10] and, typically, exhibits more spatial redundancy than texture. Depth-map quality is also less important than the quality of the synthesized views [MMS*09], created by depth-image-based rendering (DIBR) [Feh04]. Nonetheless, high-quality discontinuities are critical for avoiding displaced pixel warps, which may lead to frayed object boundaries and local inconsistencies in depth perception [VTMC11]. Since high-frequency information is typically discarded first by standard compression methods, a lossless coding is often essential, leading to a high bandwidth. Edge-preserving depth filtering [PJO*09] combined with downscaling (and upscaling while decoding) reduces bandwidth significantly [MKAM09], but it is difficult to control these filters and predict their impact.

Custom solutions Merkle et al. [MMS*09] propose *platelets*; a quadtree decomposition hierarchically dividing the depth image into blocks. Each block is split by a linear segment, defining two regions that are piecewise constant or linearly varying. The resulting sharp edges lead to higher quality than H.264/MVC that is explicitly designed for multiview video representations, but the linear edges might differ from the actual geometry, and the fitting process is relatively costly.

Template-based solutions [EWG99] potentially respect edges, but are not general enough and have difficulties in meeting bandwidth constraints. Edge detection and explicit representation results in higher precision [ZKU*04, PHE*11]. The discontinuities can be found via texture segmentation [ZKU*04] and Pająk et al. [PHE*11] show that important edges can be identified via the luminance content, but this thresholding process is overly conservative and only removes edges in low-contrast regions. While this contrast-based metric works well for their remote rendering and upsampling applications, recent findings [DRE*11, DRE*12] suggest that considering apparent depth is less than optimal, and more aggressive compression is possible.

Joint texture/depth coding Strong edges in color and depth often coincide and Wildeboer et al. [WYT*10] use the full-resolution color image to guide depth-image upsampling, similar to joint-bilateral upsampling [KCLU07]. Spatiotemporal incarnations can lead to even better results [RSD*12], or one can choose edges manually [MD08]. These approaches also improve depth scans that have notoriously low resolution and are more prone to inaccuracies than high-resolution color cameras. Unfortunately, due to variations in luminance (e.g., texture) that do not correspond to changes in depth, such reconstruction schemes may lead to so-called *texture copying* artifacts [STDT09].

In a recent compression proposal for multiview with associated depth maps (MVD) [MBM*12], depth blocks are partitioned into two constant regions separated by a straight line (similar to platelets) or an arbitrary contour. Remarkably, the partition position is derived from discontinuities in the luminance signal for corresponding blocks in the reconstructed video picture (the so-called inter-component prediction). Our approach works in the same spirit and uses color to guide depth reconstruction.

Visual masking Models of the human visual system (HVS) play an important role in improving image compression. The discrete cosine transform (DCT) image decomposition naturally approximates visual channels that are instrumental in modeling characteristics such as light adaptation, contrast sensitivity, and visual masking [Wat93]. For example, the JPEG quantization matrix elements for DCT coefficients directly model contrast sensitivity. Further, the quantization matrix can be customized per image [Wat93,RW96,SWA94] to account for visual masking, which, in compression applications, indicates how much distortion can locally be tolerated. In wavelet-based JPEG2000 [ZDL01], two important masking characteristics are modeled: (1) *self-contrast masking* where typically a compressive transducer function models increasing discrimination contrast thresholds for increasing contrast magnitude, and (2) *neighborhood masking*, where the impact of neighboring signals in space and frequency is also considered.

Visual-masking models lead to a significant bitrate reduction for color images [ZDL01, Wat93, RW96]. We explore their application in the context of disparity and depth compression. Experimental evidence [HR02, Chapter 19.6.3d] confirms the existence of disparity masking and channel-based disparity processing in the HVS. The relatively wide bandwidth of disparity channels (estimated to 2–3 octaves) suggests a less pronounced selectivity (i.e., easier modelling) and stronger impact; potentially, more spatial frequencies may mask a signal. In this work, we perform a perceptual study for compression applications, where we estimate stereoacuity in the presence of masking depth noise. We use the results to formulate a disparity masking model that is applied in our depth compression.

3. Proposed framework

State-of-the-art video coding standards, such as H.264, achieve high compression efficiency based on two important assumptions [Ric10]. First, the encoded data is assumed to come from spatially and temporally sampled *natural* (real-world) scenes. Such signals are mostly continuous in space and, due to relatively high frame rates, the change of information from one frame to the next is usually small—mostly due to camera or object motion. Video encoders exploit this spatio-temporal redundancy by only encoding data that cannot be derived or predicted from spatial or temporal neighborhoods. Second, digital videos are intended for human observers, whose visual system has performance characteristics that depend strongly on the input stimuli. Thus, frames are compressed in a perceptually linear domain, reducing information that would be invisible after decoding. Bandwidth constraints can also be better imposed in such a domain by removing features according to their visual importance. The resulting compression errors are perceptually uniformly distributed, which minimizes visibility of potential artifacts.

Unfortunately, compression tools that are designed following the above assumptions are sub-optimal for depth. Their use, even in high-bandwidth scenarios, can lead to subpar 3D video quality [DHH11]. Approaches specific to computer-graphics applications [PHE*11] can be prohibitively expensive in terms of bandwidth requirements.

In this work, instead of designing a completely custom video codec for depth, we build upon existing video encoding frameworks (Fig. 1). The high scalability and efficiency of existing codecs are maintained, and the integration of our method is simplified.

Our approach builds upon two main elements; first, a special low-resolution stream encoding that is upsampled using the separately-streamed color image (Sec. 3.2) and, second, a masking test to compress depth residuals. The masking is predicted via our model and leads to a sparse residual in the codec that then compresses more efficiently. All details about the model itself and measurements can be found in Sec. 4. We then employ this model to reduce the size of our special low-resolution stream (Sec. 4.7).

3.1. Depth Encoding

We first transform the original depth frame D into a perceptually meaningful domain. For colors, video encoders already expect data in an opponent color space. The key property of this space is a statistical decorrelation of color channels, which allows the encoder to compress them independently and focus the encoding on perceptually significant information.

In this context, we represent depth as per-pixel vergence angles. For stereo applications, where display parameters are usually known in advance, such a representation can be considered a measure of physically (and perceptually) significant depth. Therefore, we will simply refer to these values as depth throughout the paper. The depth is quantized to fixed-precision integers (Fig. 1) because the majority of current video codecs compress integer video data. Specialized solutions for floating point representations do exist, but they are less efficient in compression and run-time performance.

3.2. Color/Depth Decorrelation

To exploit the correlation between color and depth, we reconstruct the original depth D making use of two streams, a full-resolution color stream and a special low-resolution depth stream. Here, we will only concentrate on the latter and the related upsampling process.

Let D be the input depth image of resolution $d_{width} \times d_{height}$. D and the associated compressed color image C are partitioned into blocks D_i and C_i, where a block with index $i \in \{(x,y) \in \mathbb{Z}^2 : 0 \leq x < \lfloor d_{width}/k \rfloor \wedge 0 \leq y < \lfloor d_{height}/k \rfloor\}$ represents the pixels $\{k \cdot i + (x,y) : (x,y) \in \mathbb{Z}^2, 0 \leq x < k \wedge 0 \leq y < k\}$ of the image. Our approach will reconstruct D block by block.

Initially, we use a standard downsampled (by a factor k) version $\hat{d}$ of D that will be upsampled using C. Formally, $\hat{d}(i) := D(k \cdot i)$, where $I(i)$ is the pixel value of an image I at pixel position i.

To upsample $\hat{d}$ to full resolution, we rely on C and could apply one of many cross-modal upsampling techniques [KCLU07, STDT09]. However, they typically lead to severe texture copying artifacts or are computationally intensive. Hence, we propose a new fast upsampling strategy that avoids such artifacts.

Depth Upsampling A block D_i is approximated by a reconstruction D'_i, which is defined via a weighted average of four pixel values $\hat{d}$ that are in the corners of block D_i (Fig. 2). For every output pixel, the corner weights are recomputed based on weighting functions determining color similarity h_p and spatial proximity w. Using a standard spatial weighting function w (we use a simple unit "pyramid"-filter kernel, but other isotropic kernels are possible), the reconstruction is defined by:

$$D'_i(j) = \frac{\sum_{p \in \Omega_i} w((p-i) - j/k) \cdot h_p(C_i(j)) \cdot \hat{d}(p)}{w_{sum}(j)}, \qquad (1)$$

where $\Omega_i := \{i,\ i+(1,0),\ i+(0,1),\ i+(1,1)\}$ and $w_{sum}(j) := \sum_{p \in \Omega_i} w((p-i) - j/k) \cdot h_p(C_i(j))$, $j \in \{(x,y) \in \mathbb{Z} : 0 \leq x < k \wedge 0 \leq y < k\}$. The definition is similar to joint-bilateral upsampling [KCLU07], but the main difference lies in a more complex definition of the range weighting h_p, to estimate the likelihood that a given color belongs to the surface underneath p.

In the standard definition of a bilateral filter h_p would compare two pixel colors [TM98]. Nonetheless, in many cases, the color value at p can be quite different from its neighboring pixels even when they lie on the same surface. However, local image structures tend to exhibit large variations only in one specific color channel (e.g., brightness), but gradual changes in the others, producing so-called *color lines* [OW04] in color space. In our solution, we want to make use of this observation to determine a better range weighting.

Color Lines We introduce a *local color line model* (LCLM). Given a position p, we want to derive an anisotropic distance function based on the color-line orientation, which should describe the underlying surface colors better than a standard uniform distance. To this extent, we compute an approximation of a PCA of the color values in CIELAB color space in a $k \times k$-pixel neighborhood around p. Using the inverse of the eigenvalues along the axes defined by the eigenvectors leads to a distance that is virtually shrunk along the color line.

Instead of computing a full PCA, we only derive the first eigenvector of the PCA via the power iteration method. It is described by $Q_{n+1} = MQ_n/||MQ_n||$, where M is the covariance matrix and Q is the current eigenvector estimate. Three iterations are usually sufficient to find the first eigenvector, and, hence, its eigenvalue λ_p. We then define

$$\Lambda := \begin{pmatrix} \lambda_p & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_2^2 \end{pmatrix},$$

where σ_2^2 is the variance of the pixel color distances to the line defined by the eigenvector and average color μ_p.

Next, we set $T_p^{-1} := V_p \Lambda_p^{-1} V_p^T$, where V_p is an orthogonal matrix containing the first eigenvector in the first row. T_p^{-1} is then used in the definition of the weighting function h_p by involving the Mahalanobis distance

$$h_p(X) := e^{-\left(\sigma_c^{-1}\sqrt{(X-\mu_p)^T T_p^{-1}(X-\mu_p)}\right)^2},$$

where σ_c is a constant that controls the tradeoff with respect to the spatial weighting function (in practice, we use 0.1 for all examples). With this construction, we assume that color lines will indeed have a shape in color space that is rotationally invariant around the main direction, which avoids the full PCA computation and makes our approach very fast.

It is possible to increase robustness and quality even further, by excluding outliers when estimating the color line. This situation can occur when blocks contain several boundaries of significantly different surfaces. In order to do so, we use a weighted power iteration when determining Q, λ_p, and μ_p. In other words, each pixel's contribution to the color line is weighted based on its distance and similarity to p. We use the standard bilateral-filter weights, using the same weighting function w for spatial weighting and an exponential falloff with a sigma of 1.0 for range weighting. The large range sigma makes sure that only extreme outliers are excluded.

Additional Color Line Already the low-resolution depth leads to very successful reconstructions, but small features inside a block might not always be well represented by only four pixels from $\hat{d}$. These situations are easy to test for, as it means that the so-called *confidence* [DD02], i.e., sum of weights $w_{sum}(j)$ in Eq. 1, is very low.

Pixels with low confidence are unlikely to benefit from any weighted average of the depth values in $\hat{d}$. Consequently, we propose to extend the initial reconstruction formula. Our intuition is that a good reconstruction needs at least a clear separation of foreground and background. A standard downsampling will not ensure this, but if we store one additional

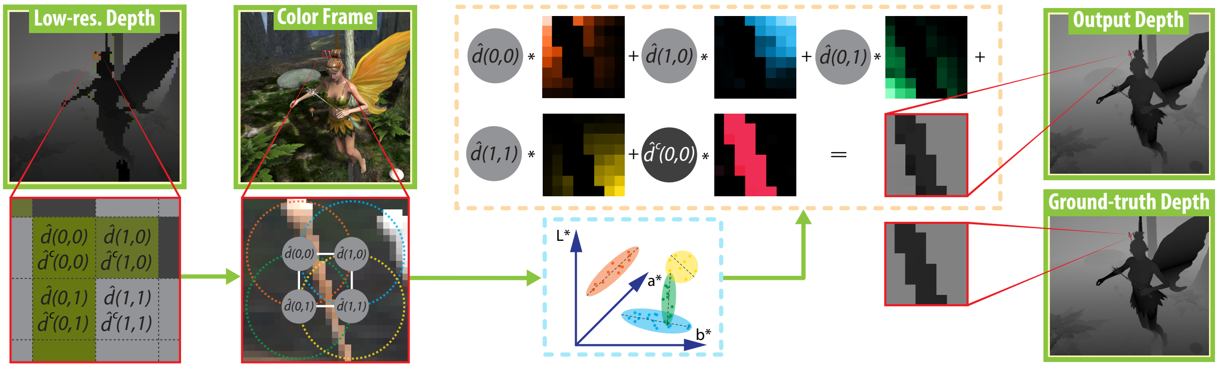
Low-res. Depth
Color Frame
$\hat{d}(0,0)$ $\hat{d}(1,0)$
$\hat{d}^c(0,0)$ $\hat{d}^c(1,0)$
$\hat{d}(0,1)$ $\hat{d}(1,1)$
$\hat{d}^c(0,1)$ $\hat{d}^c(1,1)$
d(0,0) d(1,0) d(0,1) d(1,1)
$\hat{d}(0,0)$ * + $\hat{d}(1,0)$ * + $\hat{d}(0,1)$ * +
$\hat{d}(1,1)$ * + $\hat{d}^c(0,0)$ * =
L*
a*
b*
Output Depth
Ground-truth Depth

Figure 2: *Upsampling depth via the high-resolution color frame. The color image is divided into small blocks aligned with a low-resolution depth frame. For each block we compute the first eigenvector (color line) over weighted pixel distribution in CIELAB color space. The upsampling step uses the color line model to compute distance of a pixel color to a surface described by one of the four color lines. Orange/Blue/Green/Yellow blocks visualize the product of $h_p(j)$ and $w(j)$ from Eq. 1. Additionally, subsampled geometry that has does not follow any of the four models is reconstructed with an extra color line (red block) and $\hat{d}^c$, stored in the chroma channel of the low-resolution stream.*

depth value per block and combine this step with a reconstruction making use of confidence, we can reach this goal. Using our weighted power iteration we compute a 5^{th} color line with weights $wc_i(j) = e^{-(\sigma_c^{-1} w_{sum}(j))^2}$ and modify the reconstruction:

$$D_i^c(j) = \frac{w_{sum}(j)D_i'(j) + wc_i(j)\hat{d}^c(i)}{w_{sum}(j) + wc_i(j)},$$

where $\hat{d}^c$ is an additional image stored alongside $\hat{d}$ and $wc_i(j)$ is the 5^{th} color line weight. We find the optimal value of $\hat{d}^c(i)$ by minimizing $|D_i - D_i^c|$.

3.3. Discussion of Color/Depth Decorrelation

In summary, our approach derived two one-channel images $\hat{d}$ and $\hat{d}^c$, which can be stored together in the luma and one of the chroma channels of a low-resolution image D^*. D^* is then encoded with a customized encoder, making use of our masking prediction, as explained in the next section.

In theory, we would have the possibility to store an additional value such as an additional color line in the second chroma channel, but it turned out to be of no benefit. The unused channel has no real impact on the compression quality, as simple run-length encoding (applied by any standard codec) will reduce it drastically.

It may be surprising that a low-resolution stream should be sufficient to reconstruct high-quality depth, yet it can be well motivated. For image/video coding the typical size of 8 × 8 pixels for a DCT block matches well the HVS contrast sensitivity in color vision, which roughly ranges from 1 to 50 cpd (cycles per degree). A typical display device shows signals with up to 20 cpd. Therefore, an 8 × 8 pixel DCT block is best at encoding luminance in the range of 2.5 cpd corresponding to the lowest frequency AC coefficient (half a cycle per block) to 20 cpd (highest frequency AC coefficient). However, the HVS frequency-sensitivity to depth/disparity is much lower and ranges from 0.3 cpd to 5 cpd [BR99, Fig. 1]. Therefore, an 8-pixel-wide block does not only encode very low frequencies badly, but also wastes information by encoding frequencies above 5 cpd that do not improve the perceived stereo quality. This suggests that we could downsample the depth image by a factor of $k = 4$ before encoding it, which would aim at frequencies in the range from 0.625 to 5 cpd.

However, the measurements [DRE*11] were based on frequency and ignored phase, which humans tend to be sensitive to in the presence of edges. Similar to luminance [ZDL01], more precision is needed for isolated disparity edges. Here, our upsampling is helpful, as we can recover the high-frequency depth edges from the high-resolution color frame. Furthermore, if depth edges are not matched by color edges in natural stereo images, they are also not likely to be perceived. However, this assumption does not hold for non-natural images (e.g., random dot stereograms), where no visible edge exists in the color signal, but enough luminance contrast is present to perceive depth.

4. Visual Masking for Disparity

The main goal of our experimental evaluation was to measure the effects of spatial and frequency neighborhood masking in disparity domain and to develop a perceptual model that we could later apply to depth video compression. For vergence values that change gradually, the prediction mechanisms built into H.264 work well. The residual signal R (see Fig. 1) consists of values close to zero, except for areas where prediction fails. Failures typically correspond to occlusion edges, which leads to the interesting observation that R mostly consists of edge-like structures. For this reason, we focus on measuring the discrimination thresholds between a sharp edge in disparity (difference in depth between foreground and background parts of the test stimuli) and a given random noise in disparity. In other words, we derive the minimum disparity edge magnitude that is *not* masked by the noise of a specific frequency and amplitude.

4.1. Participants

Eleven observers with an age from 21 to 44 participated in the experiment (average of 29.8 years, all males). They had normal (7 observers) or corrected to normal vision (4 wore glasses). No participants were stereo-blind. An experiment session took from 18 to 107 minutes depending on the observer. We encouraged the participants to take a break between sub-sessions to avoid visual fatigue. In a sub-session an individual pair of masker parameters were used. All participants were naïve regarding the experiment's purpose.

4.2. Apparatus

The experiments were run in two separate laboratories on two different displays: 23" LG W2363D and 23" Asus VG236H. Both are 1920 × 1080 pixel, 120 Hz displays designed to work with NVIDIA 3D Vision active shutter glasses. We set the middle grey of both displays to $\sim$40 cd/m^2 for similar brightness for the test stimuli. This initial calibration proved sufficient, as we did not find any statistically meaningful difference between the results using the different displays. The stimuli were generated in realtime on an NVIDIA Quadro graphics cards.

4.3. Stimuli

Each test stimulus consists of two superimposed signals, a masker with disparity amplitudes and frequencies changing between sub-sessions, and a horizontal edge in disparity crossing the screen through the vertical center (see Fig. 3). The edge is oriented horizontally so that both eyes can see the warped image without disocclusions. To minimize the effects of most external depth cues (e.g., shading), the stimulus is generated by warping random dot stereograms. Both signals have zero mean disparity and are located at screen depth. In order to prevent the subjects from looking for the edge in the same location, we randomize its vertical position within a 2 cm window around the screen center. The stimulus vergence map is converted into a pixel disparity map by taking into account the equipment, viewer distance (60 cm in our experiments), and screen size. We assumed standard intra-ocular distance of 65 mm, which is needed for conversion to a normalized pixel disparity over observers. To ensure sufficient quality, super-sampling was used: views were produced at 4000 × 4000 pixels, but shown as 1000 × 1000 pixel patches, downsampled using a 4 × 4 Lanczos filter.

The masker signal is computed via scale-space filtering of a white noise image such that the narrow-band result has a Gaussian distribution with mean f and standard deviation 0.5. We measured the edge discrimination thresholds for the masker frequencies centered at $f \in \{0.3,\ 0.6,\ 1.0,\ 2.0,\ 3.0\}$ [cpd], motivated by the discussion in Sec. 3.3. The masker disparity amplitude a was chosen to match the averaged noise amplitude $a(x,y) = (\sum_k |P(k)|^{0.3})^{1/0.3}$, where k is a fixed-size neighborhood around location (x,y), and P is the masker expressed in disparity [arcmin]. We tested the values $\{2.5, 5.0, 7.0, 8.0, 10.0, 16.0, 20.0\}$. The use of the 0.3-norm makes the process more robust to isolated peaks in the noise pattern (as for luminance [ZDL01]).

Experiments were conducted under dim ambient illumination. All images were static, and free eye motion allowed multiple scene fixations. Free eye fixation compensates for any pedestal disparity, guaranteeing a conservative measure of disparity sensitivity [DRE*11]. We did not account for variations in accommodation, or luminance.

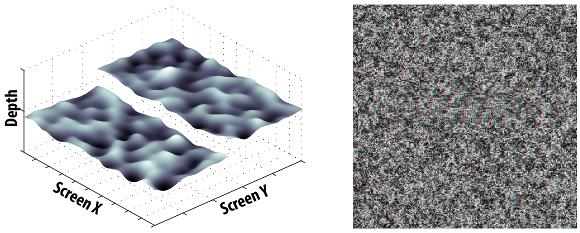

Figure 3: *Disparity edge masking experiment.* ***(left)*** *Visualization of the test stimuli—a step edge presented on top of a uniformly distributed band-pass noise in disparity. The stimuli is centered at screen depth.* ***(right)*** *The test stimuli in our experimental software application visualized in anaglyph stereo mode.*

4.4. Experimental procedure

The experiment was performed as recommended by Mantiuk et al. [MTM12]. In a training session observers familiarized themselves with the task, interface, and the shutter glasses. Then, observers were asked to look at two 3D images placed randomly either left or right on the screen (Fig. 3): both with the masker signal and one with an additional edge signal. The task was to indicate which image exhibited both signals by pressing the "left" or "right" cursor keys, meaning that they were asked to select an image in which they see a horizontal disparity edge that cuts one of the images approximately in the vertical center. In the cases where observers could not see the difference they were forced to make a random choice. After a correct/incorrect answer the discrimination-threshold Δd of the signal was decremented/incremented according to the psychometric QUEST procedure [WP83] using a Bayesian approach to estimate the likelihood of a threshold. For each given masker frequency f and masker amplitude a, the QUEST procedure was repeated until the standard deviation of the resulting probability density function (pdf) was smaller than 0.1, which took about 40 trials on average. Then, the parameters of the masking signal were changed and a new QUEST experiment was started.

4.5. Results of the experiment

In order to build a model, explaining the perception of a disparity edge profile in the presence of an additional disparity signal, all thresholds were first collected and averaged for each test stimulus (i.e., masker with specific amplitude and frequency) separately. A second-order polynomial fit to these samples resulted in the function

$$\begin{aligned}\Delta d(a,f) &= 0.9347+0.0471\cdot a+0.3047\cdot f \\ &+ 0.049\cdot a^2+0.0993\cdot a\cdot f-0.1836\cdot f^2. \end{aligned} \tag{2}$$

The value of this function (Fig. 4) represents the edge-disparity discrimination threshold for a masker defined by disparity amplitude a and its frequency f. High values indicate a strong masking effect, while lower values imply that the masker does not significantly affect the disparity-edge perception. As expected, the discrimination thresholds increase monotonically with increasing masker amplitude over the whole frequency range. The highest thresholds (i.e., the best masking properties) are obtained for maskers with high-frequency corrugations—around 0.5–1.5 cpd, depending on the masker amplitude. This observation stems from the fact that a high-frequency masker significantly reduces the visibility of the high-frequency components of the disparity step edge, which is similar to findings in luminance perception [LF80]. Due to the masking effect the high-frequency component of the signal becomes imperceptible and is no longer perceived as a step function, but instead as a noisy disparity corrugation. Hence, the model allows us to predict imperceptible edges based on the masker signal.

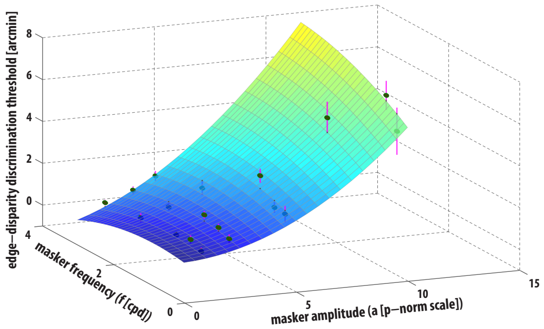

Figure 4: *Edge-disparity discrimination thresholds as a function of masker disparity frequency in [cpd] and amplitude. The function fit represented by the surface has an error of SSE = 1.5276. The grey circles show measurement locations, green circles denote mean threshold values averaged for all observers (SEM error bars are plotted). Some of the points are covered by the surface.*

4.6. Generality of the model

Our visual masking measurements were only performed for specific viewing conditions and do not directly generalize to different setups (i. e., viewing distance and display size). However, using an angular detection threshold measure makes our model independent of the screen resolution and, within the considered spatial frequency range, of the screen size. For larger screens, our model needs to rely on an extrapolated data, which can be less accurate. Some studies also indicate that our measures may stay unaffected by the viewing distance [WWP02]. For very differing setups, one could conduct additional measurements. Later, following the work on the visual difference predictors by Daly [Dal92], taking conservatively the lowest threshold across different viewing conditions would lead to a fully valid model. We envision, however, that considering a fixed setup can be greatly beneficial in the context of on-demand streaming, as specifications could be transferred by the client as part of the request to the streaming server.

4.7. Residual frame masking

Our model can be used in the compression of a depth residual R, which contains the details that are missing in the spatio-temporal prediction P produced by the codec. The idea is to set all disparities in R that are masked by P to zero, hereby optimizing the encoding of residual depth images.

Usually the residuals are compressed by various means, such as quantization, and visual masking predictions. The latter predicts if small residual values might become imperceptible when added to the predicted frame P. In this case, it is possible to set those values to zero without impacting the quality. Nevertheless, so far, codecs usually assumed color values and these models are not useful when streaming depth. With our model, we can apply visual masking for depth, which can be directly used to optimize the encoding of D^* (Fig. 5).

There are two observations to our advantage. First, R has only large values where a misprediction happened (e.g., occlusion), otherwise the values are low. Removing low values will improve compression without having a strong effect on the quality. Second, the stronger values in R mostly consist of edge-like structures (Fig. 5), which our edge-based masking model has been designed for.

The residual R produced by the codec for a depth stream D is the difference between the spatio-temporal prediction P and D. As values are given in vergence angles, their difference results in disparity. The masking threshold changes as a function of masker frequency, hence we perform a final per-pixel threshold at multiple frequencies. Following the discussion in Sec. 2, we assume that masking is performed within disparity channels whose bandwidth of 2–3 octaves is relatively large (e.g., with respect to luminance

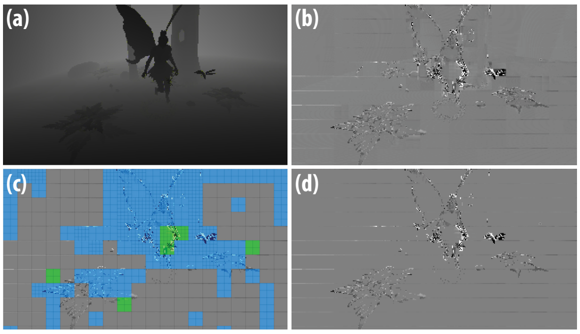

Figure 5: *Residual masking for a frame.* ***(a)*** *Input depth frame.* ***(b)*** *Frame visualized on a grey background (residual enhanced 10x for visibility).* ***(c)*** *Frame after removing masked disparities. Blue blocks have been temporarily predicted, green spatially, and grey blocks kept no residuals.* ***(d)*** *Same as (c), without colors.*

channels) [HR02, Chapter 19.6.3d]. Moreover, when encoding D^*, the predicted image P is of smaller resolution than the initial depth image D. As such, it mostly represents disparities of low and medium frequencies and allows us to limit the decomposition of P to two channels, making the procedure very efficient. The final per-pixel threshold is defined via a pooling function $\Delta d(x,y) = (\sum_i \Delta d_i(x,y)^{0.3})^{1/0.3}$ with i corresponding to the channel number. One important remark is that when checking whether P will mask R, we need to assume that the viewer is verged on P. Therefore, we first compute the (absolute) difference between P and the mean of the $k \times k$ neighborhood in P, which is conceptually equivalent to computing the amplitude of the disparity noise from our experiment. Knowing its amplitude and frequency, we can directly use the derived model to look up the discrimination threshold of a pixel from R.

5. Results

Our method is implemented in C++ and optimized via OpenMP. Performance evaluations were done on a laptop equipped with dual-core Intel Core i7-2620M. While graphics hardware could be used and is potentially faster, our CPU implementation is already very efficient. For H.264, we used an optimized *libx264* library (version 0.116) configured to encode 10-bit per color channel data in YUV444 planar format. For both H.264 and our method, the codec was set up to zero-latency (no bi-prediction) and high-quality.

Our method encodes depth quickly and respects very strict bandwidth constraints. The color-based depth upsampling is almost invariant to the subsampling ratio. The overall runtime for a 720p frame is around 18ms, out of which local color line computation took 3.5ms (for $k = 4$) or 2.5ms (for $k = 16$) and the remaining 15ms was spent on upsampling.

Two mechanisms make our approach particularly successful in a codec. First, the subsampling reduces the input video size from 16 to 256 times less pixel data, making compression faster and more effective (even motion vectors for temporal prediction become shorter and compress better). Second, during the frame encoding, our disparity masking model removes insignificant data.

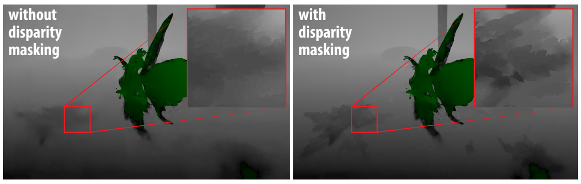

Figure 6: *H.264 compression and disparity masking.* ***(left)*** *Without masking, distortions are equally spread, but details are lost (e.g., bushes).* ***(right)*** *With masking, insignificant data is removed and bandwidth invested where useful.*

As shown in Fig. 7, our upsampling strategy results in high-quality depth maps outperforming joint-bilateral filter upsampling (JBF). For $k = 4$, both methods perform well, but for smaller geometric elements the differences become already visible. For larger k, the difference is striking. Here, more high-frequency information needs to be reconstructed via the color image. Our approach still reaches high-quality results (40dB) even for $k = 16$ and sharp boundaries, as well as smooth variations are reconstructed. The use of $\hat{d}^c$ with an additional color line is useful as illustrated by the increase in quality, despite the reconstruction already being very good. Even without it, JBF is outperformed, demonstrating how well color images can predict depth variations. Our solution is thus particularly suited for low-bandwidth environments.

Scene	Bandwidth [*Kbps*]	Resolution /subsampling	PSNR [*dB*]		
			Our w/o masking	Our with masking	H.264
BigBunny	100	1600x896/4	33.6	33.5	28.3
BreakDancers	100	1024x768/8	31.0	30.8	29.3
Fairy	100	1280x720/8	42.6	42.0	29.5
PokerTable	110	1920x1080/16	41.0	40.7	31.2

Table 1: *Quality comparison: high-res. depth reconstruction vs. our vs. H.264. Compression efficiency is tested for different scenes, resolution, texture complexity, and motion. Color/depth bandwidth was kept approximately constant – 1Mbit/100Kbps.*

In our test scenes, 100Kbps bandwidth for depth is sufficient for FullHD resolution, while standard H.264 used heavily quantized depth (the internal quantization parameter went to 63). Interestingly, the use of masking reduces PSNR (Tab. 1); the residual is necessary for a perfect frame reconstruction, but the differences are not observable. The gained bandwidth leads to significant improvements in more important regions (Fig. 6). Hence, the assumption of a uniform quantization error distribution over all pixels, from the design of H.264, is lifted and higher errors are accepted where they are masked. Numerically, the file sizes are reduced by roughly 10%, but visually the impact is much stronger.

We have decided to focus on extremely low-bandwidth

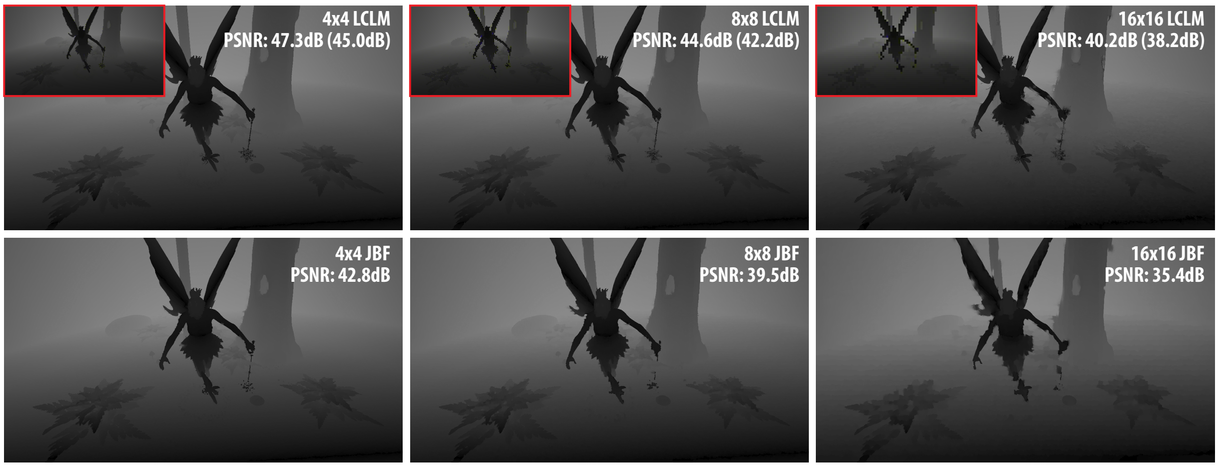

Figure 7: *Quality for various upsampling levels. Downsampled depth images are in red insets. Our (LCLM-top) vs. a joint-bilateral method (JBF-bottom) on uncompressed sparse depth. PSNR numbers in parenthesis refer to our method without* $\hat{d}^c$ *(not shown). Our quality is consistently higher. For* 4×4 *upsampling, both perform well, but JBF loses some fine geometry.*

scenarios to show that embedding 3D information in the video stream is cheap, which is one of our biggest advantages. For high-bandwidth scenarios the benefits are smaller, since MPEG codecs produce PSNR increments inversely proportional to the allocated bandwidth. In other words, the PSNR curve is logarithmic in nature. For example for $k = 4$ in the FAIRY scene, our/H.264 delivers 43.2dB/29.5dB (100Kbps), 45.9dB/40.7dB (250Kbps), 46.7dB/46.5dB (500Kbps). Note that for lossy video codecs, the 8-bit reconstruction will rarely exceed 48dB in practice. Moreover, 8-bit precision encodes the entire comfort zone (see supplementary material). In general, H.264 requires more bandwidth to obtain results close to ours in terms of PSNR, but even then, because we account for residual masking, our method encodes edges more precisely.

In this work, we compare our technique with H.264, which seems a fair choice because, of all compression techniques discussed in Sec. 2, it is the only one that is lossy, supports temporal prediction, and is equipped with a good bandwidth control mechanism. Alternatives [MWM*09] require significantly higher bandwidth, e.g., 700Kbps for the BALLETDANCERS sequence, while our approach leads to higher quality already for 100Kbps. Further, solutions such as [PHE*11] are not adequate when dealing with bandwidth limits and using 2Mbps for the FAIRY scene.

Although presented as a compression algorithm, our work has various applications. One is quality assessment, by using our model to predict visible differences, but an encoded depth map can also be used for 3D multi-viewpoint videos or personalized stereo. Also remote rendering and applications similar to those in [PHE*11] are possible.

6. Conclusion

We presented a new algorithm for color/depth encoding that outperforms previous solutions. It can be easily integrated into existing encoders, giving hope to a widespread usage. We illustrated that masking effects are very useful for depth compression and showed various applications of our model. A straightforward addition is to remove disparities from R in the regions of low luminance-contrast using the model from [CSS91], which captures a relation between disparity detection thresholds and luminance contrast.

Acknowledgements: The work was in part supported by EU FP7 project Harvest4D.

References

[BR99] BRADSHAW M. F., ROGERS B. J.: Sensitivity to horizontal and vertical corrugations defined by binocular disparity. *Vision Research 39*, 18 (1999), 3049–56. 5

[CSS91] CORMACK L. K., STEVENSON S. B., SCHOR C. M.: Interocular correlation, luminance contrast and cyclopean processing. *Vision Research 31*, 12 (1991), 2195–2207. 9

[Dal92] DALY S. J.: Visible differences predictor: an algorithm for the assessment of image fidelity. In *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology* (1992), International Society for Optics and Photonics, pp. 2–15. 7

[DD02] DURAND F., DORSEY J.: Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. on Graph. 21*, 3 (2002), 257–266. 4

[DGK*12] DOMANSKI M., GRAJEK T., KARWOWSKI D., KONIECZNY J., KURC M., LUCZAK A., RATAJCZAK R., SIAST J., STANKIEWICZ O., STANKOWSKI J., WEGNER K.: Coding of multiple video+depth using HEVC technology and reduced representations of side views and depth maps. In *Picture Coding Symposium* (2012), pp. 5–8. 2

[DHH11] DALY S., HELD R., HOFFMAN D.: Perceptual issues in stereoscopic signal processing. *IEEE Trans. on Broadcasting 57*, 2 (2011), 347 –361. 2, 3

[DRE*10] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: Adaptive image-space stereo view synthesis. In *VMV* (2010), pp. 299–306. 1

[DRE*11] DIDYK P., RITSCHEL T., EISEMANN E., MYSZKOWSKI K., SEIDEL H.-P.: A perceptual model for disparity. *ACM Trans. on Graph. 30*, 4 (2011). 2, 5, 6

[DRE*12] Didyk P., Ritschel T., Eisemann E., Myszkowski K., Seidel H.-P., Matusik W.: A luminance-contrast-aware disparity model and applications. *ACM Trans. on Graph. 31*, 6 (2012). 2

[EWG99] Eisert P., Wiegand T., Girod B.: Rate-distortion-efficient video compression using a 3D head model. In *IEEE ICIP* (1999), pp. 217–221. 2

[Feh04] Fehn C.: Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV. In *Stereoscopic Displays and Virtual Reality Systems XI* (2004), vol. 5291, SPIE, pp. 93–104. 2

[HR02] Howard I. P., Rogers B. J.: *Seeing in Depth*, vol. 2: Depth Perception. I. Porteous, Toronto, 2002. 3, 8

[HWD*09] Hewage C., Worrall S., Dogan S., Villette S., Kondoz A.: Quality evaluation of color plus depth map-based stereoscopic video. *IEEE Journal of Selected Topics in Signal Processing 3*, 2 (2009), 304–318. 1, 2

[KbCTS01] Krishnamurthy R., bing Chai B., Tao H., Sethuraman S.: Compression and transmission of depth maps for image-based rendering. In *IEEE ICIP* (2001), pp. 828–831. 2

[KCLU07] Kopf J., Cohen M. F., Lischinski D., Uyttendaele M.: Joint bilateral upsampling. *ACM Trans. on Graph. 26*, 3 (2007). 2, 4

[KHH*11] Kim C., Hornung A., Heinzle S., Matusik W., Gross M.: Multi-perspective stereoscopy from light fields. *ACM Trans. on Graph. 30*, 6 (2011), 190:1–190:10. 1

[LF80] Legge G. E., Foley J. M.: Contrast masking in human vision. *JOSA 70*, 12 (1980), 1458–1471. 7

[LHW*10] Lang M., Hornung A., Wang O., Poulakos S., Smolic A., Gross M.: Nonlinear disparity mapping for stereoscopic 3D. *ACM Trans. on Graph. 29* (2010). 1

[LIFH09] Lambooij M., IJsselsteijn W., Fortuin M., Heynderickx I.: Visual discomfort and visual fatigue of stereoscopic displays: a review. *Journal of Imaging Science and Technology 53* (2009). 1

[MBM*12] Merkle P., Bartnik C., Muller K., Marpe D., Wiegand T.: 3D video: Depth coding based on inter-component prediction of block partitions. In *Picture Coding Symposium* (2012), pp. 149–152. 2

[MD08] Maitre M., Do M.: Joint encoding of the depth image based representation using shape-adaptive wavelets. In *IEEE ICIP* (2008), pp. 1768–1771. 2

[MKAM09] Molina R., Katsaggelos A. K., Alvarez L. D., Mateos J.: Depth reconstruction filter and down/up sampling for depth coding in 3d video. *IEEE Signal Processing Letters 16*, 9 (2009), 747–750. 2

[MMS*09] Merkle P., Morvan Y., Smolic A., Farin D., Müller K., de With P. H. N., Wiegand T.: The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communcation 24*, 1-2 (2009). 2

[MTM12] Mantiuk R. K., Tomaszewska A., Mantiuk R.: Comparison of four subjective methods for image quality assessment. *Comp. Graphics Forum 31*, 8 (2012), 2478–2491. 6

[MWA*13] Masia B., Wetzstein G., Aliaga C., Raskar R., Gutierrez D.: Display adaptive 3D content remapping. *Computers & Graphics 37*, 8 (2013), 983–996. 1

[MWM*09] Merkle P., Wang Y., Muller K., Smolic A., Wiegand T.: Video plus depth compression for mobile 3D services. In *3D-TV Conference: The True Vision - Capture, Transmission and Display of 3D Video* (2009), pp. 1–4. 1, 2, 9

[OHB*11] Oskam T., Hornung A., Bowles H., Mitchell K., Gross M.: Oscam - optimized stereoscopic camera control for interactive 3D. *ACM Trans. on Graph. 30*, 6 (2011). 1

[OW04] Omer I., Werman M.: Color lines: image specific color representation. In *IEEE CVPR* (2004), vol. 2. 4

[PHE*11] Pająk D., Herzog R., Eisemann E., Myszkowski K., Seidel H.-P.: Scalable remote rendering with depth and motion-flow augmented streaming. *Comp. Graphics Forum 30*, 2 (2011), 415–424. 2, 3, 9

[PJO*09] Park Y. K., Jung K., Oh Y., Lee S., Kim J. K., Lee G., Lee H., Yun K., Hur N., Kim J.: Depth-image-based rendering for 3DTV service over T-DMB. *Signal Processing: Image Communication 24*, 1-2 (2009), 122–136. 2

[PKW11] Pece F., Kautz J., Weyrich T.: Adapting Standard Video Codecs for Depth Streaming. In *Proc. of Joint Virtual Reality Conference of EuroVR (JVRC)* (2011), pp. 59–66. 2

[Ric10] Richardson I. E.: *The H.264 Advanced Video Compression Standard*, 2nd ed. John Wiley & Sons, Ltd, 2010. 3

[RSD*12] Richardt C., Stoll C., Dodgson N. A., Seidel H.-P., Theobalt C.: Coherent spatiotemporal filterung, upsampling and rendering of rgbz videos. *Comp. Graphics Forum 31*, 2 (2012). 2

[RW96] Rosenholtz R., Watson A. B.: Perceptual adaptive JPEG coding. In *IEEE ICIP* (1996), pp. 901–904. 3

[SKHB11] Shibata T., Kim J., Hoffman D., Banks M.: The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of Vision 11*, 8 (2011). 1

[STDT09] Schuon S., Theobalt C., Davis J., Thrun S.: Lidarboost: Depth superresolution for tof 3D shape scanning. In *IEEE CVPR* (2009), pp. 343–350. 2, 4

[SWA94] Solomon J. A., Watson A. B., Ahumada A. J.: Visibility of DCT basis functions: Effects of contrast masking. In *IEEE Data Compression Conf.* (1994), pp. 361–370. 3

[TM98] Tomasi C., Manduchi R.: Bilateral filtering for gray and color images. In *ICCV* (1998), pp. 839–846. 4

[VTMC11] Vetro A., Tourapis A., Muller K., Chen T.: 3D-TV content storage and transmission. *IEEE Trans. on Broadcasting 57*, 2 (2011), 384–394. 2

[Wat93] Watson A.: DCT quantization matrices visually optimized for individual images. In *Human Vision, VisualProcessing, and Digital Display IV* (1993), pp. 202–216. 3

[WP83] Watson A., Pelli D.: Quest: A bayesian adaptive psychometric method. *Attention, Perception, & Psychophysics 33*, 2 (1983), 113–120. 6

[WWP02] Wong B. P., Woods R. L., Peli E.: Stereoacuity at distance and near. *Optometry & Vision Science 79*, 12 (2002), 771–778. 7

[WYT*10] Wildeboer M., Yendo T., Tehrani M., Fujii T., Tanimoto M.: Color based depth up-sampling for depth compression. In *Picture Coding Symposium* (2010). 2

[ZDL01] Zeng W., Daly S., Lei S.: An overview of the visual optimization tools in JPEG 2000. *Signal Processing: Image communication Journal 17*, 1 (2001), 85–104. 3, 5, 6

[ZKU*04] Zitnick C. L., Kang S. B., Uyttendaele M., Winder S., Szeliski R.: High-quality video view interpolation using a layered representation. In *Proc. of SIGGRAPH* (2004), pp. 600–608. 2