EUROGRAPHICS 2010 / T. Akenine-Möller and M. Zwicker
(Guest Editors)

Volume 29 (2010), Number 2

Perceptually-motivated Real-time Temporal Upsampling of 3D Content for High-refresh-rate Displays

Piotr Didyk[1] Elmar Eisemann[1,2] Tobias Ritschel[1] Karol Myszkowski[1] Hans-Peter Seidel[1]

[1] MPI Informatik, Saarbrücken, Germany [2] Saarland University, Saarbrücken, Germany

Abstract

High-refresh-rate displays (e. g., 120 Hz) have recently become available on the consumer market and quickly gain on popularity. One of their aims is to reduce the perceived blur created by moving objects that are tracked by the human eye. However, an improvement is only achieved if the video stream is produced at the same high refresh rate (i. e. 120 Hz). Some devices, such as LCD TVs, solve this problem by converting low-refresh-rate content (i. e. 50 Hz PAL) into a higher temporal resolution (i. e. 200 Hz) based on two-dimensional optical flow.

In our approach, we will show how rendered three-dimensional images produced by recent graphics hardware can be up-sampled more efficiently resulting in higher quality at the same time. Our algorithm relies on several perceptual findings and preserves the naturalness of the original sequence. A psychophysical study validates our approach and illustrates that temporally up-sampled video streams are preferred over the standard low-rate input by the majority of users. We show that our solution improves task performance on high-refresh-rate displays.

Categories and Subject Descriptors (according to ACM CCS): COMPUTER GRAPHICS [I.3.3]: Picture/Image Generation—Display Algorithms

1. Introduction

The continuous quest for better image quality forces display manufacturers to enhance contrast, brightness, display physical size and pixel resolution. At the same time viewers tend to move closer to the display to enjoy image details due to higher resolution and contrast, as well as to improve immersion into the visual experience, which arises from a wider field of view that is covered by the display. This has profound consequences in terms of human visual system (HVS) which creates new challenges for display technology as well. Such a wider field of view increases the role of peripheral vision, which is tuned through a specialized visual channel to low spatial frequencies and high temporal frequencies as required to detect motion (and timely react for the presence of predators) [Bur81]. This increases the viewer's sensitivity to image flickering, which becomes readily visible, in particular for bright displays. A wide field of view also increases the angular velocities of moving objects in the image, which increases perceived blur as well. This effect results from yet another visual channel in the HVS tuned to high spatial frequencies and low temporal frequencies specialized to precise object identification (e. g., food selection), which has a poor temporal response [Bur81]. Thus, to see details in a moving object its retinal image must be stabilized, which is achieved by the so-called *smooth pursuit eye motion* that tracks the moving object, so that its projection onto retina is centered in the fovea featuring the highest density of photoreceptors. Any depart of the moving object from this smooth motion trajectory is immediately perceived as image blur. Most modern displays (so-called *hold-type* displays) present moving objects at discrete positions in space for an extended period of time, thus violating the smooth motion assumption and leading to perceived blur. Common liquid crystal displays (LCD) fall into this category of hold-type displays.

An obvious way to combat both, flickering and blur, is to increase the refresh rate. Recently, 120 Hz low-cost desktop displays such as the Samsung 2233RZ and Viewsonic VX2265wm FuHzion, have been introduced. These screens can be fed externally with 120 frames per second, and do not rely on an internal frame replication as existing 100Hz+ TV sets [Kur01]. The question then arises how to efficiently synthesize frames specifically for such type of display that when investigated by a human observer lead to a sharp and convincing image.

The goal of this work is to combat flickering and blur artifacts while maintaining the original appearance of the 3D scene rendering using high-refresh-rate displays. One solution is to combat hold-type by creating many frames, but this leads to an overhead, making performance of the frame synthesis a crucial aspect. This is expected to be an even more important issue for the newly appearing Super-HD displays (4096×2160 resolution). Our solution produces additional frames at a scene independent cost, which also makes it well-suited to increase performance.

Our contribution is a temporal up-sampling scheme, tailored towards the requirements of high-refresh-rate hold-type displays and the capabilities for modern graphics hardware at the same time. We exploit the information on depth, occlusion and three-dimensional structure, made available via the GPU as well as perceptual findings to outperform pure image-based up-sampling. We diversify subsequent frames in terms of spatial frequency content to accommodate image fusion characteristics in the HVS. The required steps are simple; do not introduce lag into the video stream and can extrapolate one or even multiple frames. We perform a psychophysical study that evaluates task performance and compares perceived quality for full temporal resolution, low temporal resolution, and temporally up-sampled video streams. These advantages make our process a cheap and more-suited alternative to producing 120 Hz content directly.

This paper is organized as follows. Sections 2, 3 present selected aspects of blur perception. We review related work on hold-type blur reduction as well as temporal up-sampling required to increase image refresh rates in Section 4. Our approach is described in Section 5, and its implementation in Section 6. Results, including a perceptual study, are presented in Section 7. We conclude in Section 8.

2. Temporal Perception of Blur and Sharpness

Image sharpness is an important factor which decides upon perceived image quality [Jan01]. Perceptual studies clearly indicate [CF03, LGK06] that people prefer images with increased contrast at edges. This is often achieved by applying image sharpening techniques such as unsharp masking. Not surprisingly, blur in the image is considered an artifact and is particularly annoying when present in the regions of interest that attract visual attention such as moving objects. Blur perception is a complex phenomenon, which is affected by characteristics of the HVS such as temporal integration in the retina, eye motion, and visual illusions.

It is often assumed that *temporal integration* of information by the HVS follows Bloch's law [GT86] which states that the detectability of stimuli with similar spatial characteristics depend solely on their energy, i. e. the product of luminance and exposure time. In the context of high-refresh-rate displays, where subsequent images are fused by the HVS, this suggests that a feature displayed with enhanced intensity in a single frame is perceived in the same way as the same feature present in two frames, each with a halved intensity. We exploit this observation to maintain a local average power in each frequency component and amplify high frequencies in fully rendered frames to compensate for the blurred (warped) frames that follow. Note that Bloch's law is valid only up to some critical duration (around 40 ± 10 ms depending on spatial frequency [GT86]), which is met by high-refresh-rate displays. For longer durations only pixel luminance matters.

Motion blur naturally arises, when retinal images of objects move relatively to the retina, which may be caused by the actual object motion, eye motion, or both. Since photoreceptors in the retina integrate signal over time by an analogy to the finite exposure time in cameras, the retinal image acquired in such conditions is blurred. As we mentioned in the introduction, the *smooth pursuit eye motion* stabilizes a moving object on the fovea, which is efficient mostly for simple motions with constant velocity over predictable trajectories. However, eye tracking has its limitations [Dal98]. For low angular velocities below 0.15 deg/s the drift eye movement interferes with the smooth pursuit eye motion. Similarly, for velocities higher than 80 deg/s tracking becomes impossible.

An interesting visual illusion is the so-called *motion sharpening* [RRV74]. Surprisingly, the HVS seems to be equipped with a motion deblurring mechanism which may cause moving blurred images to appear sharper than their static counterpart. E. g., Westerink and Teunissen [WT95] have observed that for velocities higher than 15–20 deg/s the perceived sharpness of images blurred with a 6-pixel-wide filter appears similar to the original sharp images undergoing the same motion. Takeuchi and De Valois [TV05] investigated the motion sharpening effect by interleaving sharp and blurred frames. The viewers could not see any difference in the video sharpness even if two thirds of the images had been blurred, but they complained about flickering for low refresh rates. This idea has successfully been exploited in video compression and transmission applications [FB08], where selected frames have been filtered off-line to reduce the required bandwidth. In this work, we exploit these effect to propose an efficient image warping technique that enables flicker-free high-refresh-rate rendering.

3. Perception of Displays

So far, in our discussion of blur perception, we assumed that images of moving objects are perfectly reproduced in terms of motion smoothness, meaning that signal transitions at retinal photoreceptors follow real-world observation conditions.

This assumption is not valid for the existing display technology. In this section, we focus on today's predominant *hold-type* LCD displays. They exhibit two prominent forms of blur: *response time* blur and *hold-type* blur [PFD05]. Both are not present in *impulse-type* CRT displays, for which other drawbacks exist, such as flickering, lower brightness, and reduced contrast [KV04].

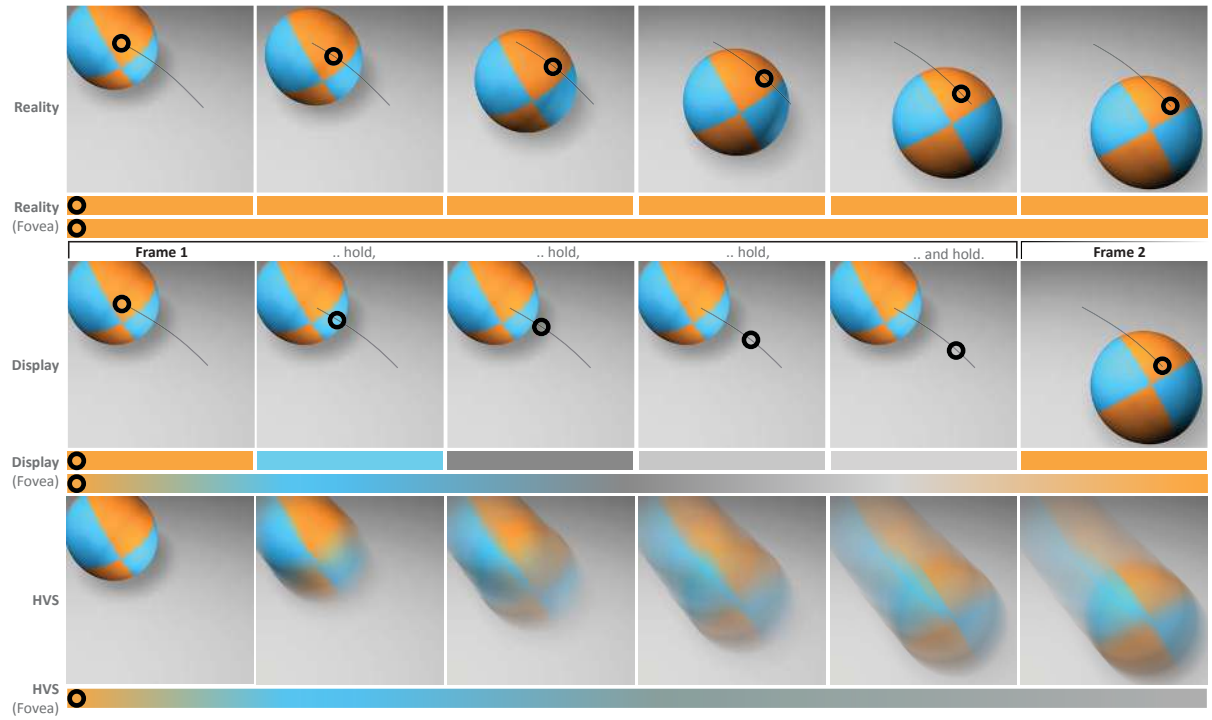

Figure 1: *A depiction of hold-type blur for a ball moving with a translational motion of constant velocity. In the top row we show six intermediate positions at equal time intervals taken from a continuous motion. The empty circles denote the eye fixation point resulting from a continuous smooth-pursuit eye motion that tracks some region of interest. For each instance of time, the same relative point on the ball is projected to the same location in the fovea, which results in a blur-free retinal image. The central row shows the corresponding hold-type display situation. Here, the continuous motion is captured only at the two extreme positions. Frame 1 is shown during a finite amount of time, while the eye fixation point follows the same path as in the top row. This time, different image regions are projected to the same point on the retina. Temporal integration registers an average color leading to perceived blur as shown in the bottom row.*

Response time blur results from the inability of the LCD display to switch between intensity levels instantaneously, but its contribution to the overall blur is relatively low. Pan et al. [PFD05] report that only 30 % of blur is a consequence of the response time, even for slow 16 ms displays. For modern displays, the response time of 3–4 ms becomes negligible and *overdrive* algorithms can even lead to further reductions [Fen06].

Hold-type blur is a purely perceptual effect arising from an interaction between the HVS and hold-type displays [PFD05]. The blur is not physically present in the image and cannot be measured with e. g., a high-speed camera. Hold-type blur can be seen as the inverse of motion blur: In motion blur, the eye is fixed and an object moves, leading to blur, while for the hold-type effect, the image is held constant while the eye moves. Fig. 1 explains the mechanisms causing the hold-type blur.

Hold-type blur can be modeled as a convolution with a box filter oriented in the object motion direction: As the eye moves, the retinal projection moves and, therefore, is spread across a constant box-shaped profile [KV04]. The box-filter support size, and thus the strength of blur, depends on the moving pattern, velocity, and the frame-hold duration. The faster the motion, the longer the distance in terms of pixels and consequently, this leads to more blur. With increasing refresh rates, the hold-type blur is reduced because the hold time itself is getting shorter. Fig. 2 illustrates the amount of perceived blur, introduced by hold-type displays when reducing the refresh rate from 120 Hz to 60 Hz. In the limit, smaller hold times can remove hold-type blur completely, but this requires feeding displays at higher frame rates.

4. Previous Work

This section reviews related work on hold-type blur compensation and temporal up-sampling because it represents the basis of our work.

In 3D rendering, motion blur can be introduced explicitly for artistic reasons or to mimic the finite exposure time of real-world cameras. In this paper, we do not discuss synthesized

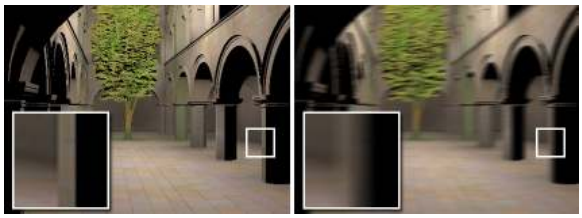

Figure 2: *Simulation of hold-type blur. An animation sequence with the sample frame as shown on the left is displayed simultaneously with 60 and 120 Hz refresh rate on a Samsung 2233RZ 120 Hz display. The effective velocity of horizontal motion as seen on the screen is the same in both cases. The user's task is to adjust the blur in the sequence refreshed with 120 Hz until the level of blur matches the 60 Hz sequence. The average outcome of such an experiment is shown on the right. In other words, the sequence of blurred frames (right) at 120 Hz are visually equivalent to the sequence of sharp frames (left) displayed at 60 Hz.*

blur and refer the interested reader to the extensive survey by Sung et al. [SPW02].

Hold-Type Blur Compensation Many modern TV sets feature increased refresh rates such as 100 and 200 Hz (respectively 120 and, 240 Hz for the NTSC standard) as a means to reduce hold-type blur. Intermediate frames are added internally relying only on a standard (low-framerate) broadcasting signal. Feng et al. [FPD08] provide an extensive survey of existing methods for hold-type blur reduction.

The simplest option is *black data insertion* (BDI) which means interleaving original frames with black frames to reduce the hold time. This comes at the expense of flickering (best visible in bright flat regions), brightness reduction, and color desaturation. Original frames can also be duplicated by adding blurred copies [CKL*05], but this causes visible ghosting as no motion compensation is performed.

Backlight flashing (BF) [PFD05, Fen06] is an efficient alternative and overcomes the LC response time problems by flashing the backlight (500 and 600 Hz TVs sets are available). In modern devices, backlights are built out of hundreds of LEDs, that are flashed only after the LCD reaches its target level. Although helpful for response-time blur, this approach is prone to visible flickering and reduces brightness due to shorter backlight duty cycles. Note that BF and BDI essentially mimic impulse-type displays, such as CRT. They, hence, reintroduce drawbacks of older displays.

Frame rate doubling by interpolation (FRT) [Kur01] derives in-between frames. Each pair of original frames is interpolated along the optical flow trajectories.

Optical flow is a difficult problem, prone to artifacts which affect the quality of in-between frames. Our experimental investigations, which we conducted on a modern TV set using a high speed camera (1,200 frames-per-second), revealed that such algorithms tend to fail for occlusions, high velocity motion, and highly textured regions. To avoid visible errors optical flow is automatically deactivated in case of doubts and original frames are simply replicated, at expense of increasing hold-type blur. Even such precautions do not help in all situations and objectionable artifacts still can appear for some realistic scenarios. In the executable accompanying to this paper we show a pathological case, which cannot be handled by the tested TV set. In this respect our technique is quite general and additionally benefits from high-quality motion flow derived from 3D scene and camera data.

Another drawback is, that interpolation naturally results in a time lag which is not a problem for broadcasting applications, but cannot necessarily be tolerated for games or other interactive applications. If an input arrives between two frames (no matter the display frequency), the interaction is visualized only in the next frame. Thus, 60 Hz react with 30 Hz in the worst case. Perceptual experiments showed that subjects could detect delays in the interaction even beyond 90 Hz [LWC*02]. Interestingly, it is sometimes stated that beyond 60 Hz, no performance increase is possible [LWC*02]. This depends strongly on the task and display. Our study shows that in dynamic environments higher refresh rates do have an important impact. In fact, temporal visual lags can be perceived as a strong distraction for some cross-modalities, as studied for audio [DS80], haptics [LMM*00] or physics [OD01].

Motion-compensated inverse filtering (MCIF) [KV04] is a software alternative to suppress hold-type blur. Motion vectors are locally computed using a space-time recursive search blockmatcher. It is then assumed that eye tracking locally follows these motion vectors and each frame is sharpened using a local 1D filter, which is oriented along the motion direction. The sharpening strength is chosen to compensate for hold-type blur. The effectiveness of this method is limited by the fact that hold-type blur removes highest frequencies irrespectively of their enhanced (sharpened) contrast. Also, the amount of sharpening that is required for perfect blur compensation would lead to extreme filter band-pass properties, which is not feasible due to possible intensity clipping.

TV set manufacturers also use hybrid solutions that combine BF and FRT at higher refresh rates, but details on such custom solutions are not published. Note that our approach could be combined with selected techniques such as BF.

Temporal Up-Sampling Displaying more frames is one option to remove hold-type blur. Generating these frames relates to morphing, a classic computer graphics problem, see Wolberg's survey [Wol98]. We morph frames using 3D information generated as a by-product of GPU rendering and exploit perceptual findings to compensate for inaccuracies.

Stich et al. [SLW*08] address perceptual effects in temporal up-sampling of image sequences. They show that high-quality moving edges are a key feature to the HVS and that

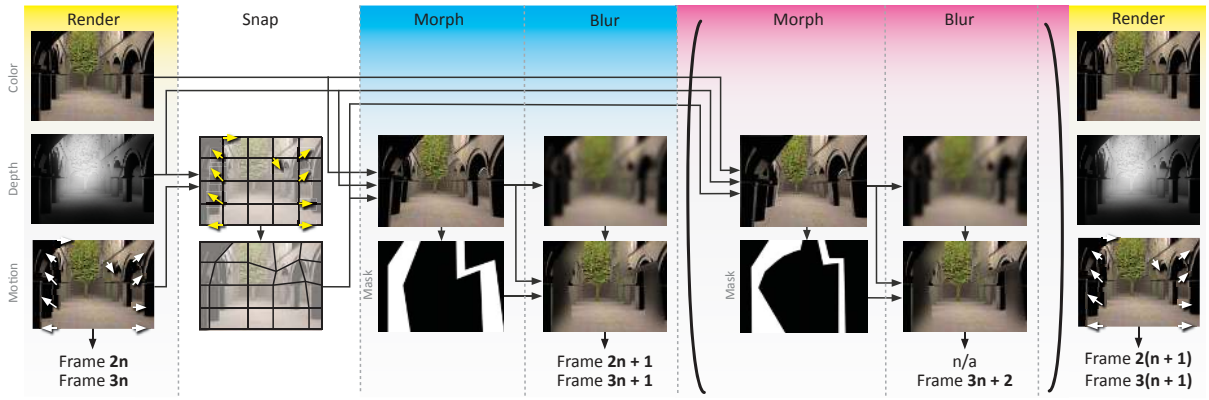

Figure 3: *Our pipeline, from left to right: To extrapolate one (or multiple) in-between frames, we use motion flow to warp the previously shaded result into an in-between frame, that is then locally blurred to hide artifacts caused by morphing failures. Finally, we compensate for the lost high-frequencies due to this blur by adding additional high frequencies where necessary.*

ghosting, and comprised edge sharpness, both arising from hold-type blur, can be a strong distraction. They improve the perceived edge quality by making their movement more coherent over time via interpolation.

Liu et al. [LGJA09] use content-preserving warps to stabilize video. Stabilized frames are warped such that the original image content remains intact. We can rely on blur instead of content-preserving warps. The latter are computationally expensive and not required at high frame rates.

Up-sampling for videos has recently been addressed by Mahajan et al. [MHM*09]. Their work is very well-suited for a single disocclusion, but it requires full knowledge of future frames and can be computationally expensive (reporting several orders of magnitude more compute power).

As mentioned before, these interpolations introduce a lag, which can be an issue for reactivity. Another advantage is, that our up-sampling operator takes a few milliseconds, making it suitable for real-time purposes, whereas the other here-mentioned methods target offline video.

In the context of 3D interactive applications, up-sampling has been addressed by Mark et al. [MMB97]. They re-project shaded pixels exploiting information from the depth buffer. Such re-use of shaded samples is also the basis of the Render Cache [WDP99] which is effective, e. g, in global illumination where samples are very expensive. More recently Nahab et al. [NSL*07] proposed a new caching scheme, exchanging forward for reverse mapping.

5. Our Approach

In this section we give a high-level introduction to our up-sampling scheme, while its implementation is detailed in Section 6. Our approach is effective, improving quality by using 3D information (i. e. occlusions) and by exploiting the limitations of the HVS (i. e. insertion of blur at high frequencies is not detectable).

5.1. Pipeline

To reduce hold-type blur we propose to up-sample the stream of rendered images using a pipeline as depicted in Fig. 3. Information extracted during rendering allows us to improve the quality of in-between frames and accelerate their computation. To extrapolate one (or multiple) in-between frames, we use motion flow to warp the previously shaded result that is then blurred [CKL*05] where required. The additional blur allows us to hide artifacts if morphing fails and makes extrapolation sufficiently accurate. The lost high frequencies are compensated for by adding additional high frequencies where needed.

Motion flow Contrary to motion flow from videos, we can extract motion flow during rendering. The graphics card has knowledge about object displacements, which is different from special displays to combat hold-type blur because they need to reverse engineer imperfect 3D motion via optical flow. By taking the difference in position for every vertex, we can compute perfect motion flow and rasterize it into a buffer. While higher-order motion models are possible, a linear assumption proved sufficient in our tests.

Morphing Morphing takes the original frame and maps every pixel into its new predicted position, but this can be costly. In our implementation, we make this mapping piecewise linear by mapping a subset of pixels – a *grid* – and extrapolating the deformation over this grid.

Morphing might map multiple source pixels to a single destination pixel. We can resolve such ambiguities, by relying on depth, extracted just like the motion flow. Note that such information is not available to image-based approaches such as those used by TV manufacturers.

We will show that blur can remove inconsistencies to a large extent, but morphing a fixed resolution regular grid can lead to significant artifacts that are not easily fixable.

E. g., diagonal edges, or discontinuities can fall between grid vertices, such as depicted in Fig. 4 (left). Our solution to this problem is to snap the grid to deformation discontinuities (i. e. optical flow) in the *original* domain Fig. 4 (middle), before morphing them to their new location in the *morphed* domain as seen in Fig. 4 (right).

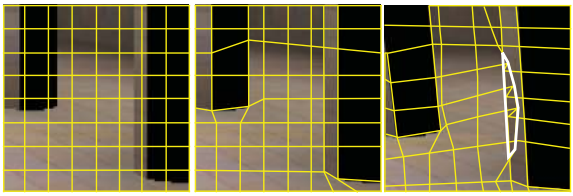

Figure 4: *To warp the original (Left) into the in-between frame (Right), we proceed in two steps. First, a uniform grid (Left) is snapped to discontinuities in motion flow (Middle). Second, those vertices are warped into a new location (Right). By doing so, discontinuities in motion are preserved, which is an important perceptual cue. Further, conventional depth buffering resolves overlaps (White area) by comparing depth values from the original frame to a depth buffer in the target frame before writing.*

Preferably the snapping is done to nearby edges, hereby trading of regularity against adaptivity. Vertices on the image border, are kept at their location in order to prevent undefined regions, e. g., black borders. Note, that this warping does not lead to disocclusion holes, as is usually the case for reprojection. This makes special hole-filling strategies unnecessary.

Blur Morphing can result in artifacts, because, even though we handle new occlusions using depth values, disocclusions remain a challenging problem. Disoccluded surfaces are not present in the original frame and selective re-rendering is expensive for rasterization, even when using masking techniques. Especially, the entire geometry would need to be processed again.

Fortunately, the possibility to interleave high-frequency and low-frequency content allows us to improve upon these problems. As small features that appear due to disocclusion would result in high-frequency content, consequently, by blurring the image with a Gaussian kernel, we hide potentially missing information.

The downside is that such an operation changes the frequency content of the image and we need to compensate by adding back high frequencies. This is difficult for in-between frames due to the lack of information, but it can be done for the original image by subtracting a blurred version. Exploiting the HVS incapability to detect interleaved high- and low-frequency content at high refresh-rates allows us to produce a visually equivalent output by adding increased high frequencies to the original frame only.

Gamma Correction We need to ensure that the increased frequencies lead to the correct appearance when integrated over time by the HVS. In theory, if we use $N-1$ in-between frames, it is sufficient to scale the high-frequency layer by a factor of N before adding it back. In practice, the process is slightly more involved because we need to also counteract the display's gamma curve. For this derivation, we will assume that the image is stationary, and we denote the high- and low-frequency layer H, L, respectively, γ the gamma exponent of the display, and $\hat{H}$ the modified detail layer needed to ensure a correct compensation for the blurred in-between frames. Over N frames the result should, energy-wise, be equivalent to $N(H+L)^\gamma$. For our $N-1$ in-between frames, we have $(N-1)L^\gamma$.

$$\begin{aligned} N(H+L)^\gamma = \;& (L+\hat{H})^\gamma + (N-1)L^\gamma \qquad \leftrightarrow \\ \hat{H} = \;& (N(H+L)^\gamma - (N-1)L^\gamma)^{\frac{1}{\gamma}} - L \end{aligned}$$

Only for $\gamma = 1$, we get $\hat{H} = NH$ (a simple scaling).

Selective Blur Although it is in theory desirable to apply the blur to the entire image [CKL*05], some problems can make it preferable to apply the blur selectively.

Such a selective blur poses two problems. First, the modified original frame can saturate and exceed the display's dynamic range. Second, we might not always be able to reproduce perfect black levels when relying on blurred frames. Because the blur makes neighboring pixels bleed into black areas, these black pixels can contain grey values in the blurred frames that make the black pixels appear slightly brighter. Although these two problems might at first glance not be related, both are a consequence of physical limitations. For saturation, we exceed the upper bound of displayable brightness, and to compensate for the brightening of black pixels, we would need to be able to display negative values in the enhanced original frame.

To address these issues, we perform a simple analysis after having split the frame in its low- and high-frequency content, as shown in Fig. 5. We verify whether we cross the boundaries of the displayable dynamic range and, if this is the case, we will reintroduce some of the high-frequency content in the low-frequency layer. If the original pixel is darker than the low-frequency counterpart, we keep the original. If the enhanced original exceeds the limits of the dynamic range, we subtract the exceeded content and shift it to the low-frequency layer. Energy-wise and, thus, integrated over time, these operations deliver the correct result. It might seem necessary to propagate the locations of such modifications to the following in-between frames in order to correctly compensate for these changes, but it is unnecessary because decisions are based on luminance values only.

For disocclusions and strong deformations, the warped grid content cannot be reliable and any high-frequency information increases the likelihood of artifacts. Thus, we maintain these regions blurred and use a smooth-step function that

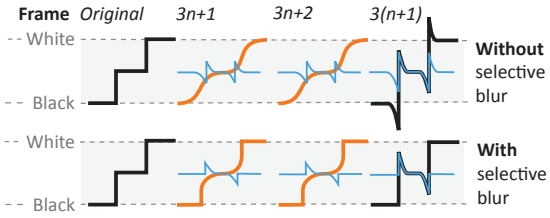

Figure 5: *Selective blur: The first row presents the situation, where blur is introduced every frame and full compensation is not possible due to limited dynamic range. The second row depicts, how the problem is solved by shifting some content from the original frame into the blurred frames.*

maps grid distortion to blur strength, to blend between the true low-frequency content and our enhanced version.

5.2. Limitations

Some limitations for this approach have to be kept in mind.

Motion Flow Pixels affected by transparency (e. g., transparent materials, simulated motion blur or depth of field), do not have a simple motion flow. The mapping of such pixels to multiple motion flows and the introduction of strategies for specular materials (e. g., glass), or meshes with changing topology are left for future work.

Morphing We assume a certain predictability and linearity of motion flow over the in-between frame. Consequently, discontinuities in velocity, might not be well represented, leading to motion that smears over the in-between frame. Our blur somewhat counteracts this phenomenon and the potential problem was not observed in practice, even when the actual motion is highly non-linear (e. g., rotating fan). For more irregular motion this problem is further reduced because of limited tracking capabilities of human observers in such scenarios.

6. Implementation

Our up-sampling is implemented in vertex and fragment shaders. While current GPUs are very fast, it is still challenging to perform frame extrapolation in a few milliseconds. Therefore we describe implementation details in this section.

Morphing We morph frames by warping a two-dimensional distorted (snapped) grid of $N \times M$ vertices. To respect discontinuities we want to translate each vertex from its regular grid position to a nearby discontinuity (maximal gradient) in the motion flow. For this, each vertex examines a small neighborhood around its original position (typically 8×8). To avoid snapping two vertices to the same location, we choose the original grid such that no two neighborhoods overlap, but the entire image is covered.

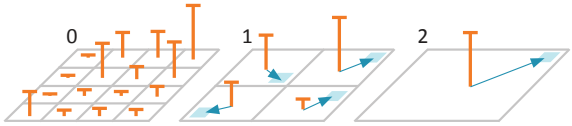

Figure 6: *Finding the maximum gradient in a neighborhood: At level 0 of a 4×4 grid, different gradients are denoted as vertical bars. Going to level 1, the maximum gradient (vertical bar), as well as a pointer (blue arrow) to the location of the maximum (blue square) is stored on a 2×2 grid. Level 2 stores the maximum and its location.*

We find the maximum value and its location by relying on a special form of MIP map (cf. Figure 6). For level 0, each pixel stores the gradient magnitude and its coordinates. For successive levels i, we recursively combine four pixels from level $i-1$. We find the maximum gradient value and copy the entire entry to level $i-1$. The result is a traditional max MIP map that additionally stores *where* the maximum occurred. In practice, we encode a relative position with respect to the vertex that will search the corresponding neighborhood. This allows us to quantize the information, 2×5 bits for position, and 6 bits for gradient magnitude, in a total of 16 bits per pixels. This fine-grained parallel strategy leads to a speedup of a factor of two over a sequential loop in the vertex program to find the maximum.

After finding the maximum, we snap the vertex to this location and adjust its texture coordinates to reflect the new position. In this way, the grid is warped and respects discontinuities, but the texture is still undistorted. The distortion only comes from the motion flow, which is then additionally applied to the vertex position.

While we allow (and intend) fold-overs, we still draw a closed, connected grid of $N \times M$ vertices from an OpenGL vertex buffer (2×16 bit per vertex) to achieve a $N-1 \times M-1$ tile grid. Further, we enable the depth test and pass depth from the original frame to resolve occlusions at fold-overs.

Blur Instead of employing a full gaussian blur, we use a MIP map with a recursive 3-tap binomial weight filter. We then read the MIP map at a higher level using tri-cubic reconstruction. As the blur occurs after tone mapping, it is done using 8 bit RGB values.

Motion Flow To compute high-quality per-pixel motion flow, each vertex' position is transformed into homogeneous clip space at time t and time $t+\Delta t$. During rasterization, the two homogeneous vertex positions are projected into Euclidean space and their difference produces the optical flow for that pixel which avoids problems at the clipping planes.

Extrapolating frames using previous frame motion flow for high velocities and complex motion is difficult. Fortunately, the tracking performance of human observers is limited and

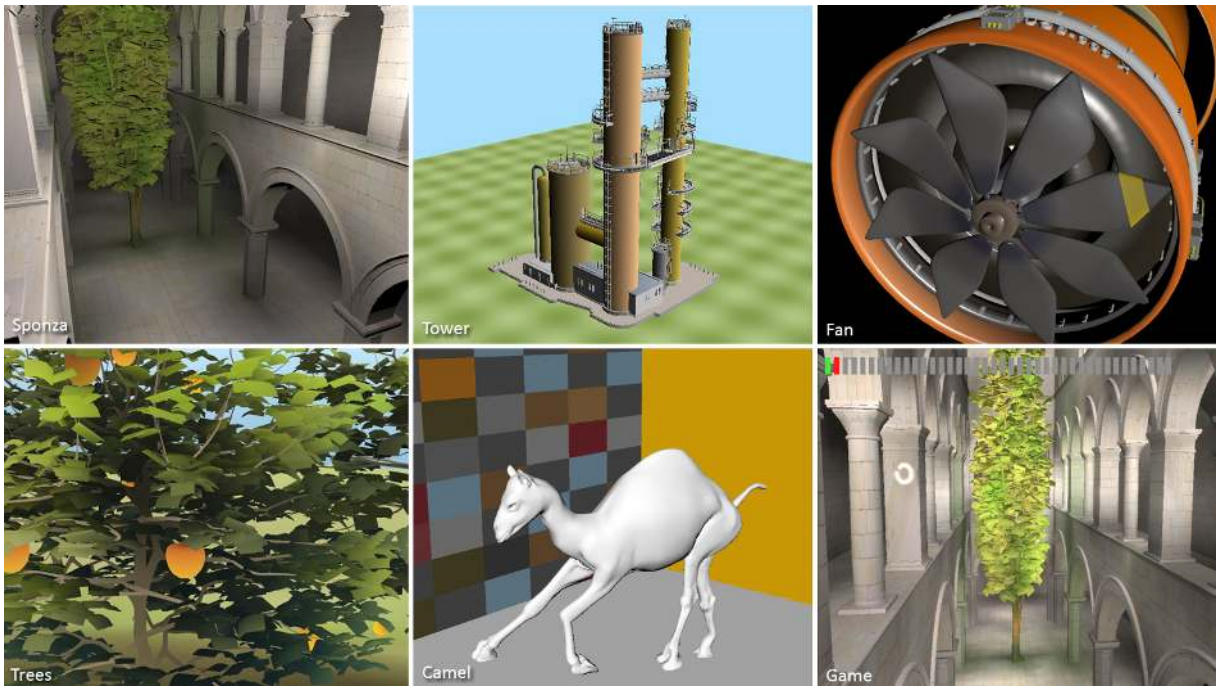

Figure 7: *The stimuli used. The „Sponza" scene has moderate geometric and texture detail. „Tower" has many occlusions and disocclusions are difficult to extrapolate in image space. „Fan" shows rotational movement that is difficult to extrapolate. Even more heterogeneous movement is found in the „Camel" mesh animation. Many occlusions and disocclusions occur in the „Tree" scene. The „Game" scene was used to measure task performance.*

allows us to bound the deformations. Based on the findings in [Dal98] tracking is possible up to 80 deg/s.

According to these findings, we simulate the loss of tracking accuracy by a simple function f. For 70 deg/s we assume perfect tracking ($f(70) := 1$), for 90 deg/s no tracking ($f(90) := 0$). Using a cubic smooth-step curve ($f'(70) := f'(90) := 0$) gives good overall results. We extend beyond 80 deg/s because Dali et al. measured random motion, whereas 3D scenes usually exhibit more coherence. In fact, velocity damping is usually preferred, even over 120 Hz (see next Section), as it tends to reduce blur (in this case the motion blur due to imperfect tracking).

7. Results

7.1. Performance

Table 1 presents performance numbers for our technique on an 3.0 GHz Core 2 Duo CPU with an NVIDIA GTX 260.

Scene	Motion Flow	Morph	Blur	Total
Sponza	0.40 ms	1.92 ms	3.34 ms	5.66 ms
Tower	1.64 ms	1.95 ms	3.36 ms	6.95 ms
Fan	0.33 ms	1.86 ms	3.38 ms	5.57 ms
Trees	1.00 ms	1.93 ms	3.38 ms	6.31 ms
Camel	0.49 ms	1.75 ms	3.37 ms	5.61 ms

Table 1: *Performance breakdown for various scenes when upsampling 40 Hz to 120 Hz (resolution is* 1024×1024*). If rendering takes more than half of the total upsampling time to produce one frame, our operator is useful as it produces two frames at the same time.*

7.2. Experimental Validation

We conducted a series of psychophysical experiments to understand how our temporal upscaling compares to standard rendering methods with respect to blur-reduction, possibly introduced artifacts, and game-related task performance. In this section, we outline only major findings of our experiments, while details concerning the study design, participants, apparatus, and statistical data analysis are provided as supplemental material.

Rating The goal of the first experiment was to judge the amount of perceived blur by rating three rendering methods: our temporal upsampling from 40 Hz to 120 Hz, and native rendering with low (40 Hz) and high (120 Hz) framerate. All three pre-rendered sequences have been simultaneously shown on a 120 Hz Samsung SyncMaster 2233 next to each other in a randomized order. 14 subjects had unlimited time, during which ≈ 20 s long sequences were looped, to rate the perceived amount of blur in the scale from 1 to 9 for each

rendering method. The stimuli depicted in Figure 7, covering a range of possible applications, such as computer games or medical and technical visualization have been used (refer also to the accompanying video). We diversified also stimuli in terms motion complexity, which decides upon the eye tracking efficiency.

Fig. 8 summarizes the obtained results. Independent ANOVA tests computed for each stimuli revealed statistically meaningful differences in the perceived amount of blur between rendering methods (for detailed statistics refer to the supplemental material). Adjusted pair wise contrasts (the paired sample t-test with the Bonferroni correction) indicate that for all scenes (except „Camel", which we included to the study as the special case) our method performed significantly better with respect to native 40 Hz rendering and comparably to 120 Hz rendering (in the latter case a statistically significant difference has only been found for the „Sponza" scene).

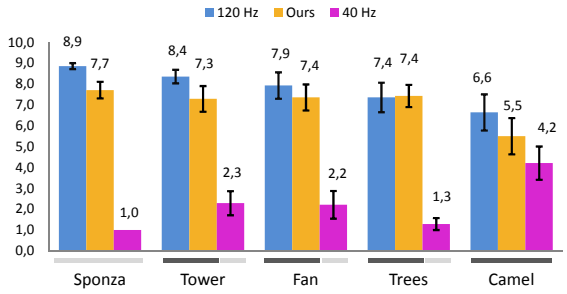

Figure 8: *Quality rating for 5 scenes. The dark horizontal lines under each scene indicate* no significant statistical difference *(series of t-test). Error bars represent* ± 1 *SEM (standard error of the mean).*

We included one special scene („Camel") where it is virtually impossible for an observer to track the fast and complicated leg motion. In this case no hold-type blur is present and all three tested methods performed similarly. Our goal was to show that our method is failsafe for untrackable motion, as it locally reduces morphing based on a prediction of poor eye tracking (refer to Section 6). Our rendering outcome is perceived comparable to native 40 Hz rendering, whereas 120 Hz, due to the lack of tracking, results in perceived distinct copies of legs at discrete positions (like a strobing effect in undersampled motion blur rendering [SPW02]), which even reduces the overall contrast. To investigate this case further, we informally asked 10 subjects to report on similarity in the appearance of our and 120 Hz sequences with respect to a selected static frame. The subjects reported better match in similarity for our method. This observation may suggest that brute-force increasing of the framerate may not always improve the animation appearance, and local frame processing that anticipates the eye-tracking ability is required.

Artifacts The next important question is whether our method does introduce artifacts as a side effect of blur reduction. In a second experiment that immediately followed the first one, the subjects were presented the same animation sequences again, but this time our method was singled out by a red frame. The subjects were asked whether they see any artifacts in our sequence which they cannot see or are much weaker in the other two sequences. By asking this specific question and giving unlimited time for the answer we wanted to ensure that the subjects carefully analyze the presence of possible artifacts. The side-by-side comparison eases the detection of differences significantly. Further, we did not specify any kind of possible artifacts to not bias the subjects in their observations. The vast majority did not report any observations for any of the sequences (over 82 % responses). Apart from isolated remarks on the differences in shadows (justified), contrast and color changes (the latter two, mentioned in 3 % of the cases, seem to be less grounded), all other comments addressed various aspects of temporal aliasing. The subjects reported that such artifacts, due to undersampling, are slightly more pronounced in our rendering with respect to 120 Hz sequences. Temporal aliasing has been mostly reported (in all but one cases) for „Sponza", „Tower", and „Trees" scenes, where the camera is panning and natural supersampling of pixels fused by the eye is achieved for 120 Hz rendering. Perhaps, this effect can explain the slightly lower rating of our method with respect to 120 Hz as can be seen in Fig. 8, although aliasing was not directly related to hold-blur rating in this experiment. Similar observations were not made in the context of 40 Hz rendering, probably due to the excessive hold-type blur.

We conclude from those findings that our temporal upsampling is comparable to 120 Hz rendering in terms hold-type blur reduction and overall animation appearance, but with significantly less computational effort. Our technique does not cause additional aliasing with respect to 40 Hz rendering and the slight difference to 120 Hz was only seen by a few subjects.

Game We finally demonstrate that our approach can lead to a better task performance by a simple game (refer to the „Game" scene in Figure 7), in which the participant is asked to tell apart two classes of moving targets. We use a three-dimensional Landolt circles as target classes, which we show in a randomized fashion and ask the participant to press one button when a target is a closed circle or a different button if it is an open circle. Not pressing a button with an object in sight is counted as failure. Pressing a button without an object in sight is ignored. We investigated four rendering scenarios: native rendering with refresh rate of 40 Hz, 60 Hz, and 120 Hz, as well as our temporal upsampling from 40 Hz to 120 Hz. 10 subjects took part in the experiment. On average the scores obtained by the subjects playing using our method were 45 % better than those for original 40 Hz, 12.7 % better than for 60 Hz and 3.3 % worse than for 120 Hz. The statistical analysis with ANOVA over the scores for each method reveals the main effect ($F(3,27) = 17.07, p < 0.00001$). Adjusted

pair wise contrasts (the paired sample t-test with the Bonferroni adjustment) indicate statistically significant differences between our approach and 40 Hz ($t(9) = 7.71, p < 0.001$) as well as 60 Hz ($t(9) = 4.25, p < 0.01$). No effect has been found when our technique has been compared to 120 Hz ($t(9) = -2.18, p > 0.05$). We conclude, that the hold-type blur effect can decrease task performance while our approach can restore it to a quantifiable extend.

8. Conclusions

In this paper, we presented an efficient GPU-based up-sampling approach that can reduce hold-type blur significantly for 3D content such as video games or animations, as shown in our study. Implementing our solution in a small hardware device is an interesting avenue of future work. Alternatively, our technique can optionally use a secondary and cheaper GPU to perform the up-sampling task. Combinations of temporal with spatial up-sampling or spatial super-resolution are worth investigating.

Our morphing method could also be beneficial in the context of stereo vision. Instead of synthesizing a second view explicitly, it could instead be produced via the morphing process.

Acknowledgements We would like to thank Gernot Ziegler and David Luebke of NVIDIA corporation for providing a Samsung SyncMaster 2233 RZ display as well as Matthias Ihrke for helpful comments on the design of our user study. This work was partially supported by the Cluster of Excellence MMCI (www.m2ci.org).

References

[Bur81] BURR D.: Temporal summation of moving images by the human visual system. In *Proc. of the Royal Society of London* (1981), vol. B 211, pp. 321–339. 1

[CF03] CALABRIA A., FAIRCHILD M.: Perceived image contrast and observer preference I: The effects of lightness, chroma, and sharpness manipulations on contrast perception. *J Imag. Sci. & Tech. 47* (2003), 479–493. 2

[CKL*05] CHEN H., KIM S.-S., LEE S.-H., KWON O.-J., SUNG J.-H.: Nonlinearity compensated smooth frame insertion for motion-blur reduction in LCD. In *Proc. Multimedia Signal Processing, 2005 IEEE 7th Workshop on* (2005), pp. 1–4. 4, 5, 6

[Dal98] DALY S.: Engineering observations from spatiovelocity and spatiotemporal visual models. In *Human Vision and Electronic Imaging III* (1998), SPIE Vol. 3299, pp. 180–191. 2, 8

[DS80] DIXON N. F., SPITZ L.: The detection of auditory visual desynchrony. *Perception 9*, 6 (1980). 4

[FB08] FUJIBAYASHI A., BOON C. S.: A masking model for motion sharpening phenomenon in video sequences. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E91-A*, 6 (2008), 1408–1415. 2

[Fen06] FENG X.-F.: LCD motion blur analysis, perception, and reduction using synchronized backlight flashing. In *Human Vision and Electronic Imaging XI* (2006), SPIE Vol. 6057, pp. M1–14. 3, 4

[FPD08] FENG X.-F., PAN H., DALY S.: Comparisons of motion-blur assessment strategies for newly emergent LCD and backlight driving technologies. *Journal of the Society for Information Display 16* (2008), 981–988. 4

[GT86] GOREA A., TYLER C. W.: New look at Bloch's law for contrast. *Journal of the Optical Society of America A 3*, 1 (1986), 52–61. 2

[Jan01] JANSSEN R.: *Computational Image Quality*. Spie Press, Bellingham, Washington USA, 2001. 2

[Kur01] KURITA T.: Moving picture quality improvement for hold-type AM-LCDs. In *Society for Information Display (SID) '01* (2001), pp. 986–989. 1, 4

[KV04] KLOMPENHOUWER M. A., VELTHOVEN L. J.: Motion blur reduction for liquid crystal displays: Motion-compensated inverse filtering. In *Proc. SPIE, Vol. 5308, 690* (2004). 2, 3, 4

[LGJA09] LIU F., GLEICHER M., JIN H., AGARWALA A.: Content-preserving warps for 3D video stabilization. *ACM Trans. Graph (Proc. SIGGRAPH) 28* (2009). 5

[LGK06] LIN W., GAI Y., KASSIM A.: Perceptual impact of edge sharpness in images. *Vision, Image and Signal Processing, IEE Proceedings 152*, 2 (April 2006), 215–223. 2

[LMM*00] LEVITIN D. J., MACLEAN K., MATHEWS M., CHU L., JENSEN E.: The perception of cross-modal simultaneity. In *Proc. CASYS* (2000), pp. 323–329. 4

[LWC*02] LUEBKE D., WATSON B., COHEN J. D., REDDY M., VARSHNEY A.: *Level of Detail for 3D Graphics*. Elsevier Science Inc., New York, NY, USA, 2002. 4

[MHM*09] MAHAJAN D., HUANG F.-C., MATUSIK W., RAMAMOORTHI R., BELHUMEUR P.: Moving gradients: A path-based method for plausible image interpolation. *ACM Trans. Graph. (Proc. SIGGRAPH '09) 28*, 3 (2009). 5

[MMB97] MARK W. R., MCMILLAN L., BISHOP G.: Post-rendering 3D warping. In *Proc. ACM I3D* (1997), pp. 7–16. 5

[NSL*07] NEHAB D. F., SANDER P. V., LAWRENCE J., TATARCHUK N., ISIDORO J.: Accelerating real-time shading with reverse reprojection caching. In *Graphics Hardware* (2007), pp. 25–35. 5

[OD01] O'SULLIVAN C., DINGLIANA J.: Collisions and perception. *ACM Trans. Graph. 20* (2001), 151–168. 4

[PFD05] PAN H., FENG X.-F., DALY S.: LCD motion blur modeling and analysis. In *Proc. ICIP* (2005), pp. 21–24. 2, 3, 4

[RRV74] RAMACHANDRAN V. S., RAO V. M., VIDYASAGAR T. R.: Sharpness constancy during movement perception (short note). *Perception 3*, 1 (1974), 97–98. 2

[SLW*08] STICH T., LINZ C., WALLRAVEN C., CUNNINGHAM D., MAGNOR M.: Perception-motivated interpolation of image sequences. In *Proc. APGV* (2008), pp. 97–106. 4

[SPW02] SUNG K., PEARCE A., WANG C.: Spatial-temporal antialiasing. *IEEE Transactions on Visualization and Computer Graphics 8*, 2 (2002), 144–153. 4, 9

[TV05] TAKEUCHI T., VALOIS K. D.: Sharpening image motion based on the spatio-temporal characteristics of human vision. In *Proc. SPIE, Vol. 5666, 690* (2005), pp. 83–94. 2

[WDP99] WALTER B., DRETTAKIS G., PARKER S.: Interactive rendering using render cache. In *Proc. EGSR* (1999), pp. 19–30. 5

[Wol98] WOLBERG G.: Image morphing: A survey. *The Visual Computer 14*, 8 (1998). 4

[WT95] WESTERINK J., TEUNISSEN C.: Perceived sharpness in complex moving images. *Displays 16*, 2 (1995), 89–96. 2