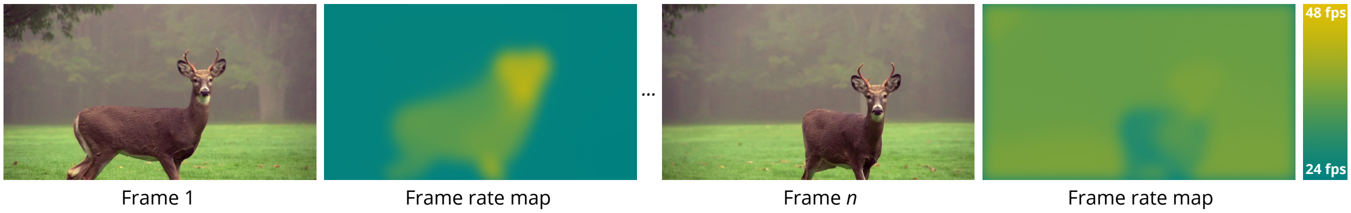
Emulating Displays with Continuously Varying Frame Rates

Krzysztof Templin[1] Piotr Didyk[2,1] Karol Myszkowski[1] Hans-Peter Seidel[1]

[1]MPI Informatik [2]Saarland University, MMCI



Frame 1 Frame rate map Frame *n* Frame rate map

Figure 1: *Using different presentation frame rates yields different looks of the motion picture: higher rates reduce visibility of artifacts such as strobing and judder, whereas lower rates contribute to the "cinematic look" of the film. We introduce a technique that enables emulating the look of any presentation frame rate up to the display system frame rate. The frame rate in the content processed with our method can vary continuously, in both the spatial and the temporal dimensions.* Deer *sequence: (CC) Jeffrey Beach*

Abstract

The visual quality of a motion picture is significantly influenced by the choice of the presentation frame rate. Increasing the frame rate improves the clarity of the image and helps to alleviate many artifacts, such as blur, strobing, flicker, or judder. These benefits, however, come at the price of losing well-established film aesthetics, often referred to as the "cinematic look". Current technology leaves artists with a sparse set of choices, e.g., 24 Hz or 48 Hz, limiting the freedom in adjusting the frame rate to artistic needs, content, and display technology. In this paper, we solve this problem by proposing a novel filtering technique which enables emulating the whole spectrum of presentation frame rates on a single-frame-rate display. The key component of our technique is a set of simple yet powerful filters calibrated and evaluated in psychophysical experiments. By varying their parameters we can achieve an impression of continuously varying presentation frame rate in both the spatial and temporal dimensions. This allows artists to achieve the best balance between the aesthetics and the objective quality of the motion picture. Furthermore, we show how our technique, informed by cinematic guidelines, can adapt to the content and achieve this balance automatically.

Keywords: presentation frame rate, cinema, HFR, strobing, judder

Concepts: •**Computing methodologies** → *Image processing; Perception;*

1 Introduction

With the recent release of Peter Jackson's *Hobbit* trilogy in the HFR (high frame rate) format, another attempt was made to break with the almost century-old tradition of shooting films at 24 frames per second. It has been announced that Andy Serkis' *Animal Farm* and the sequels of James Cameron's *Avatar* will also employ high frame rates; thus, one can already talk about an emerging trend in film-making, which is backed by the presence of temporal up-sampling capabilities in most modern home entertainment systems. Increasing the acquisition and presentation frame rate helps to alleviate many artifacts of motion pictures, such as blur, strobing, flicker, or double edges, and thus leads to a more faithful image reproduction. These artifacts, however, contribute to the well-established aesthetics of the film, and the reactions of the audiences to the increased frame rate have been mixed so far. Many commentators contrast the classic "other-worldly, cinematic look" of 24-fps motion pictures with the "cheap, soap-opera look" of films presented at higher frame rates. This is a paradoxical situation in which improving the objective reproduction quality leads to an inferior subjective experience. At the same time, many people prefer the cleaner look of high frame rates, and a well-grounded argument has been put forward that increasing the frame rate helps to minimize the visual discomfort experienced during stereoscopic viewing.

It seems that high frame rates work better for some types of content than others (e.g., documentaries, sports events) or even certain types of shots within a single film (e.g., establishing shots). The choice of the frame rate, therefore, could be seen as a creative decision, and it was suggested that variable frame rates should be employed, so that the artist can select on a case-by-case basis the frame rate that best serves the storytelling purpose [Quesnel et al. 2013]. Solutions combining two different frame rates have been proposed; however, they still give a rather limited control over the look of the film. In their short film *Lucid Dreams of Gabriel*, Disney Research [2015] demonstrated how to embed lower-frame-rate content within a higher-frame-rate sequence (6 fps and 24 fps within 48 fps). It remains unclear, however, how to embed content whose frame rate is not a divisor of the higher frame rate without introducing video stutter. Similarly, Trumbull and Jackson [2013] discuss only certain combinations of frame rates, without the possibility to vary the frame rate continuously. Due to this limited choice of frame rate pairs, in certain situations either the film aesthetics or its objective quality has to be compromised.

SIGGRAPH 2016 Technical Paper, July 24–28, 2016, Anaheim, CA
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925879

To solve this problem, we introduce a technique that, given a standard, single-frame-rate display, enables emulation of the whole spectrum of lower presentation frame rates without introducing low-frequency judder. The novelty of our approach lies also in the ability to smoothly vary the frame rate in both the spatial and the temporal dimensions (see Fig. 1). This gives artists more creative freedom and enables them to achieve the best balance between the aesthetics and the quality of the motion picture. Our technique can operate automatically by analyzing the optic flow in the scene and locally adjusting the frame rate based on cinematic guidelines. Our paper makes the following contributions:

- a technique for continuous interpolation between different frame rates,
- a perceptual study calibrating the interpolation so that it can emulate different presentation frame rates,
- evaluation of the proposed technique,
- an automatic application locally minimizing the emulated presentation frame rate for optimized cinematic effect.

2 Related Work

The appearance of motion in a video is strongly dependent on the characteristics of the employed camera and display. Due to the inherently discrete nature of the acquisition and presentation process, distortions of the motion often become apparent. In this section, we review the most common motion reproduction artifacts and discuss specific camera and display designs that try to actively influence the appearance of motion.

2.1 Motion Reproduction Artifacts

Differences between the real-world motion and the sampled motion can be treated as a distortion signal, whose spatial and temporal properties determine its detectability by the visual system [Daly 1998]. Watson [2013] presented a theoretical framework for distortion detection, accounting for virtually all important characteristics of modern imaging systems. However, it is not clear how to account for the distortion magnitude and correlate it with the appearance of different distortion manifestations, such as motion blur, strobing, repeated edges, or flickering, since they are often hard to isolate. Daly et al. [2014] and Hoffman et al. [2011] measured the visibility of such artifacts in extensive psychophysical experiments, and they found out that artifacts become more visible with decreasing frame rate as well as with increasing velocity and image contrast. Since the contrast sensitivity changes with the base luminance, the image brightness can potentially affect the artifact visibility [Hoffman et al. 2011]. However, this was not observed for typical display brightness levels [Daly et al. 2014]. In what follows, we briefly characterize all these artifacts and the ways one can influence their appearance.

Motion Blur Due to the use of rotary disc shutters in motion picture cameras, the exposure time in cinematography is often referred to as the "shutter angle", and measured in degrees, with 360° corresponding to the time that elapses between two consecutive exposures, 180° corresponding to half that time, 90° to one fourth, etc. The shutter angle of 180° is widely considered standard, and for a typical 24-fps film this means 1/48 s of exposure. Motion blur, the result of extended exposure time, is commonly used in motion picture photography to reduce spatio-temporal aliasing or to meet certain artistic goals, and many rendering systems simulate finite exposure time to provide similar capabilities [Sung et al. 2002]. Emulation of arbitrary shutter angles can be relatively easy achieved in a post-process by blending consecutive frames interpolated along optical flow trajectories [Brostow and Essa 2001]. The perception of motion blur is highly non-linear. Daly et al. [2014] observed a clear motion artifact reduction for the shutter angles over 90°. This observation is consistent with the outcome of motion blur matching experiment by Navarro et al. [2011], where a high number of incorrect matches was found for shutter angles below 180°. This was attributed to strong strobing artifacts that dominated the blur appearance. Motion sharpening effects have been observed for blurred patterns moving with high velocity [Westerink and Teunissen 1995].

Repeated Edges Repeated edges (edge banding, ghosting) appear when the frame rate is insufficient to reproduce fast motion [Watson 2013], or too-short exposure times are used [Stengel et al. 2015]. Edge banding is reduced when motion is actively tracked by the eyes, although this observation does not hold in the case of repeated frame flashing (as in cinema projectors), when more banding can be observed for the tracked object [Hoffman et al. 2011]. Interestingly, eye motion has little influence on the visibility of other artifacts such as flicker and strobing [Hoffman et al. 2011]. The ghosting visibility can be reduced by increasing motion blur or decreasing contrast of moving objects, but the approach that is the least invasive in terms of the image appearance is to increase the frame rate. This effectively reduces gaps between edge duplicates and in the limit leads to their fusion into a less objectionable blurred percept. Obviously, the same effect can be achieved by reducing the object speed, and in particular the visibility of banding artifacts does not change significantly when the ratio between the frame rate and velocity is constant [Hoffman et al. 2011]. Ghosting incarnations unrelated to object motion can also arise for certain rendering techniques, e.g., blending of different views in image-based rendering [Vangorp et al. 2011] or cross-talk between views in stereoscopic and light-field displays [Zwicker et al. 2006; Chai et al. 2000]. The strong negative impact of ghosting on the image quality is consistently stressed in all discussed scenarios.

Strobing Short exposures result in strobing, i.e., discontinuous, stuttered motion, which can be reduced by increasing motion blur [Navarro et al. 2011]. In particular, strong strobing effects can be seen at high-contrast features during fast panning of the camera at low frame rates [Stengel et al. 2015].

Flickering Flickering perception is often characterized by the *critical flicker frequency* (CFF), which is the frame rate that is required for steady perception (fusion) of temporally changing patterns [Kalloniatis and Luu 2009]. The CFF increases with contrast of flickering patterns as well as with display brightness (the Ferry-Porter law) [de Lange 1958]. The spatial extent of the flickering pattern affects the CFF, which may drop even below 40 Hz for stimuli smaller than 0.3° of visual angle [Mäkelä et al. 1994]. The flicker visibility is reduced in hold-type displays or those with repeated frame flashing [Watson 2013].

With the trend toward larger and brighter displays as well as faster moving images, all discussed distortions will become even more noticeable [Daly et al. 2014]. Nevertheless, a controlled mixture of such distortions may result in a desirable motion appearance akin to the "cinematographic look". While modeling interactions of such distortions and independent control of their contributions to the cinematographic look seems to be an exciting goal [Daly et al. 2014], in this work we introduce a video manipulation technique, that influences the artifact visibility indirectly, via emulation of different presentation frame rates. Because of the difficulties in independent analysis of each discussed distortion, in this work we distinguish only between the distortions of temporal nature (such as strobing and flicker), which we collectively denote as *judder*, and their spatial counterparts akin to blur.

2.2 Temporal Shaping for Cameras and Displays

Cameras Motion picture acquisition can be seen as a temporal sampling process, with the pre-filtering kernel size determined by the shutter angle, i.e., the exposure time. Instead of using a continuous kernel, a frame can be assembled from a series of aligned short exposures to control the amount of motion blur [Telleen et al. 2007]. Raskar et al. [2006] used a fluttering high-speed, ferro-electric LCD shutter to restore the high-frequency content in dynamic scenes. The Time Filter system [Tessive 2014] uses steerable a liquid-crystal filter synchronized with the camera shutter to smooth temporal integration kernels, which results in reduced temporal aliasing. The newer, digital version of this system, Time Shaper, captures 120 frames per second with a nearly-360° shutter and integrates them in a post-process. This way, kernels with sizes greater than 360° and negative weights (required for theoretically optimal filtering) can be simulated. Fuchs et al. [2010] pursued this approach for video, and processed content captured with a high-speed camera (500 fps) to investigate different reconstruction filters for mapping the input stream to 60 fps output. In their work, temporal aliasing properties were analyzed and several editing tools suggested. The main focus in the above lines of work was on the theoretical properties of the sampling scheme, with the goal of reducing aliasing, controlling the motion blur, and developing new cinematic looks. In contrast, our goal is to closely emulate the appearance of the whole range of presentation rates and enable fine-scale control over the final outcome.

Displays Displays present moving objects at discontinuous (“frozen”) positions, and when the eyes track the motion, the so-called hold-type blur arises. In contrast to motion blur, it is formed in the visual system; however, both types of blur have a very similar appearance [Feng 2006]. One can increase the magnitude of high frequencies in the image, but the full compensation for their suppression by hold-type blur might not be possible due to a limited image dynamic range. Hold-type blur is suppressed by reducing the time span when each frame is displayed. This can be naturally achieved by some form of temporal up-sampling, e.g., by interpolating between the existing frames, but also by black-frame insertion or backlight flashing [Stich et al. 2008; Didyk et al. 2010].

Cinema Before the introduction of sound films, around which time the standard of 24 fps was born, films were captured and projected at various frame rates. Sixteen frames per second was considered standard, but rates much lower as well as much higher than that were not uncommon, with some productions combining several rates within one show [Brownlow 1980]. In the early 1980s, Douglas Trumbull developed the Showscan system running medium-format film at 60 fps, which gave the audience an experience of extremely high temporal and spatial resolution. In his experiments, increasing the frame rate amplified the emotional response in the audience. The new embodiment of these ideas – the Showscan Digital system – captures images at 120 fps using a nearly-360° shutter. This allows for integration of the frames, effectively simulating acquisition at several lower rates [Stump 2014, p. 133]. The proposed system is complemented by the functionality to automatically combine two frame rates within one scene, depending on the pixel luminance temporal variation [Trumbull and Jackson 2013]. We extend this idea to emulate non-discrete presentation frame rate selection and smooth rate variation across space and time. A very recent study by Wilcox et al. [2015] shows, that people generally prefer higher frame rates. However, the study focused on image quality (motion smoothness, clarity, etc.), and did not investigate artistic aspects of frame rate variation (“film look”). Moreover, only stereoscopic 3D content was used, but this medium seems to be less compatible with techniques that intentionally reduce image quality.

3 Overview

In this paper, we show how given a single-frame-rate display one can interpolate between different frame rates and emulate lower presentation frame rates via image temporal filtering. Here, by “emulation” we mean that the actual frame rate of the video stream and of the display are equal and constant over time; however, the frames of the video are synthesized in such a way, that a *perception* of any desired lower presentation frame rate is obtained. For example, given a 48 Hz display and a desired frame rate of 36 fps, our technique produces a 48-fps sequence that, when shown on the 48-Hz display, closely resembles the same content recorded at 36 fps and shown on a 36-Hz display. Note that displaying 36-fps content directly on a 48-Hz display requires doubling of some frames and leads to a low-frequency stutter.

We derive our technique in two steps: First, we show how to smoothly interpolate between different presentation frame rates using our *kernel displacement* technique (Sec. 4). Second, we perform a perceptual experiment in which we find the correspondence between the interpolation parameters and veridical frame rates (Sec. 5). This enables the emulation of arbitrary lower presentation frame rates on a single frame rate display.

4 Presentation Frame Rate Interpolation

Before we describe our technique for interpolating between different frame rates, we formally introduce the concepts of motion picture *acquisition* and *presentation*, which we use to explain our method.

4.1 Motion Picture Acquisition

The acquisition (i.e., sampling) of a given motion picture frame can be modeled as a convolution of a continuous, time-dependent signal S with a rectangular filter. The temporal support of the filter is proportional to normalized shutter $w = \alpha/360^\circ$ and inversely proportional to frame rate f, and is defined as:

$$\mathrm{rect}_{f,w}(t) = \begin{cases} f/w & \text{when } |t| < w/(2f), \\ 0 & \text{otherwise.} \end{cases}$$

The temporal sampling positions are always distributed uniformly: for a given frame rate f, the sampling time of frame I_k is described by function $\mathcal{T}_f(k)\colon \mathbb{N}\to\mathbb{R}$, $\mathcal{T}_f(k) = t_0 + k/f$, where t_0 is the sampling time of I_0. Using the above definitions, the sampled frame sequence is given by:

$$I_k = \int_{-\infty}^{\infty} S(t) \cdot \mathrm{rect}_{f,w}\left(t - \mathcal{T}_f(k)\right) \mathrm{d}t.$$

An illustration of such a sampling for a 180°-shutter camera is provided in Fig. 2a.

4.2 Motion Picture Presentation

Given a display which operates at f frames per second, a sequence corresponding to the signal S sampled at rate f can be presented directly. It is also straightforward to present content at frame rates lower than f that result from dividing the presentation frame rate by a positive integer (i.e., $f/2$, $f/3$, $f/4$, ...). To this end, it is enough to repeat every frame a fixed number of times, which formally means

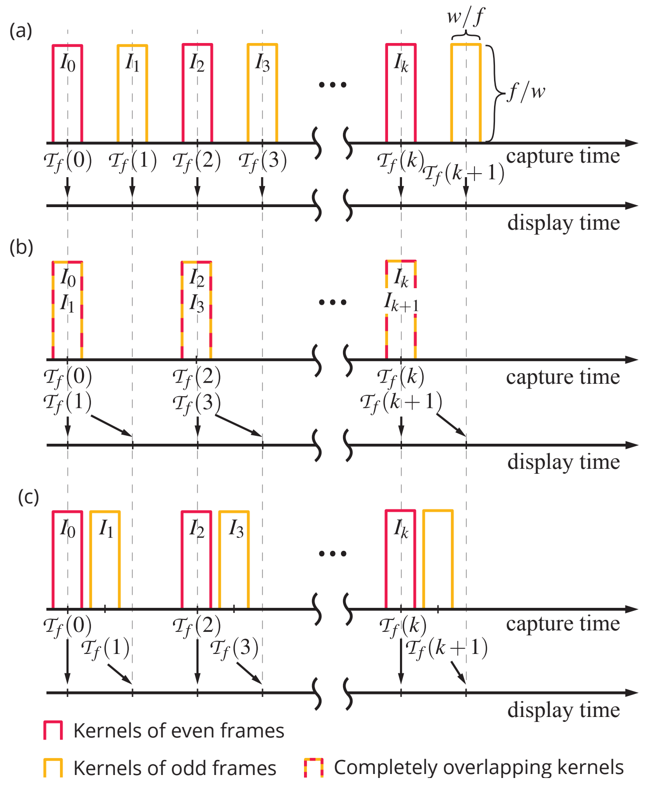

(a)
w/f
f/w
I_0
I_1
I_2
I_3
I_k
$\mathcal{T}_f(0)$ $\mathcal{T}_f(1)$ $\mathcal{T}_f(2)$ $\mathcal{T}_f(3)$ $\mathcal{T}_f(k)$ $\mathcal{T}_f(k+1)$
capture time
display time
(b)
I_0 I_1
I_2 I_3
I_k I_{k+1}
$\mathcal{T}_f(0)$ $\mathcal{T}_f(1)$ $\mathcal{T}_f(2)$ $\mathcal{T}_f(3)$ $\mathcal{T}_f(k)$ $\mathcal{T}_f(k+1)$
capture time
display time
(c)
I_0 I_1 I_2 I_3 I_k
$\mathcal{T}_f(0)$ $\mathcal{T}_f(1)$ $\mathcal{T}_f(2)$ $\mathcal{T}_f(3)$ $\mathcal{T}_f(k)$ $\mathcal{T}_f(k+1)$
capture time
display time
Kernels of even frames
Kernels of odd frames
Completely overlapping kernels

Figure 2: (a) *The sampling kernels of an f-fps film captured with the standard* 180° *shutter.* (b) *A straightforward emulation of an* $(f/2)$*-fps,* 90° *shutter sequence – the sampling positions of odd display frames are equal to those of even display frames. As a result, the display behaves like an* $(f/2)$*-fps one, while still operating at f frames per second.* (c) *In order to emulate in-between frame rates we interpolate the extreme situations from* (a) *and* (b)*, which is achieved via kernel displacement, resulting in their uneven spacing. For clarity, we focus on the kernel placement here; kernel size is discussed in Fig. 3. At this point, the kernel distribution in* (c) *does not correspond to any specific frame rate; the perceptual effect of this operation is studied in later sections. Note that the positions of kernels correspond to the sampling time, not to the time when they are actually displayed. The presentation time is always the same and is fully determined by the display system.*

that for a number of consecutive frames the sampling position of signal S does not change. For instance, to emulate the $(f/2)$-fps rate every sampling position is used twice, which corresponds to the following modification of $\mathcal{T}_f$:

$$\mathcal{T}'_f(k) = \begin{cases} t_0 + k/f & \text{for even } k, \\ t_0 + (k-1)/f & \text{for odd } k. \end{cases}$$

Note that this leads to a situation in which the acquisition times of odd frames do not exactly correspond to their presentation times (see Fig. 2b for an illustration). As a result of this modified sampling, the display – nominally still operating at f frames per second – emulates an $(f/2)$ Hz display. This is an exact emulation, since the obtained output either closely matches or is equivalent to what would be seen if a real $(f/2)$ Hz display and camera were used. In a similar fashion, one can achieve even lower frame rates by modifying the number of times each sampling position is repeated.

The above example is a special case of the more general solution that repeats some – but not all – sampling positions. Such a technique can be used to emulate arbitrary frame rates, and in fact, it is routinely used by most video players, which repeat certain frames when required to play content of a lower frame rate on a display with a higher frame rate. This approach, however, introduces additional, unwanted temporal frequencies, causing non-smooth motion (video stutter), which is easily spotted by the observer. For example, one can emulate a 40-fps display at the 48-fps playback rate by repeating every fifth sampling position, but this results in objectionable 8 Hz stutter. Moreover, when one tries to use this method to spatially interpolate between different frame rates, a moiré pattern appears. We provide an illustration of these artifacts in the supplementary video.

4.3 Smooth Frame Rate Interpolation

We propose a technique that overcomes the above limitations and enables emulation of arbitrary frame rates below the display frame rate. An important feature of our solution is that the frame rate can be smoothly varied over the spatial and temporal domains without introducing visible artifacts. For clarity of exposition, we describe how to interpolate between $f/2$ and f frames per second, where f is the display frame rate, and we discuss the generalization of the technique to lower frame rates in Sec. 4.4.

Our key observation is that the difference between the extreme cases of f fps and $f/2$ fps is the *position* of the odd sampling kernels (Figs. 2a and 2b). To achieve smooth interpolation between these two situations, we *displace* kernels of the odd frames to locations between the two positions corresponding to $f/2$ and f fps (Fig. 2c). This operation can be defined using a new function $\mathcal{T}_f^\delta$, $\delta \in [0,1]$, interpolating between the original $\mathcal{T}_f$ and its modified version $\mathcal{T}'_f$:

$$\mathcal{T}_f^\delta(k) = \begin{cases} t_0 + k/f & \text{for even } k, \\ t_0 + (k-\delta)/f & \text{for odd } k. \end{cases}$$

Note that $\delta = 0$ and $\delta = 1$ provide the sampling for the f-fps and the $(f/2)$-fps case, respectively, i.e., $\mathcal{T}_f^0 \equiv \mathcal{T}_f$ and $\mathcal{T}_f^1 \equiv \mathcal{T}'_f$.

Although displacing kernel positions interpolates between two frame rates, the exposure time in terms of the shutter angle is not preserved, because the kernels do not change their width. To solve this problem, we also interpolate the width of sampling kernels using a generalized version of the sampling function:

$$\mathrm{rect}_{f,w}^\gamma(t) = \begin{cases} (1-\gamma/2)f/w & \text{when } |t| < w/((2-\gamma)f), \\ 0 & \text{otherwise,} \end{cases}$$

where $\gamma \in [0,1]$ is an interpolation parameter. The full interpolation including both sampling positions *and* the kernel size is illustrated in Fig. 3. An example of a corresponding image sequence is shown in Fig. 4.

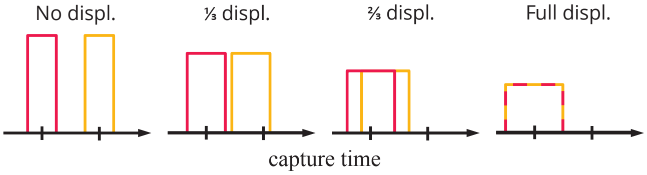


Figure 3: *Interpolation between f fps,* 180° *and $f/2$ fps,* 180°*. From left to right: no displacement, one-third displacement, two-thirds displacement, and full displacement. Since the shutter angle is constant, the absolute exposure time at both ends is different, and it needs to be smoothly interpolated along with the kernel position.*

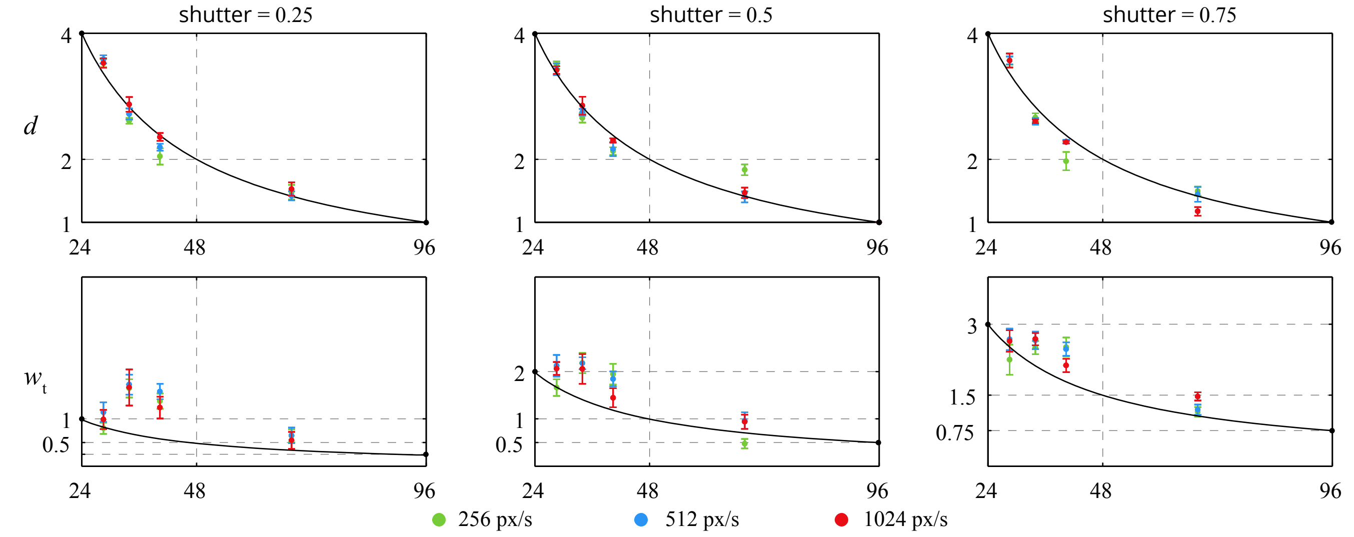


Figure 5: *Results of the calibration experiment. Each point is the average of responses of 10 subjects, and the error bars are the standard errors of the mean. The upper row corresponds to the displacement parameter d, and the lower row to the shutter angle parameter w_t. The black solid lines in the upper row indicate the displacement proportional to the inverse of the frame rate. The solid lines in the lower row indicate constant absolute exposure time (the w_t value is always relative to the frame rate of 96 fps). See the main text for a discussion.*

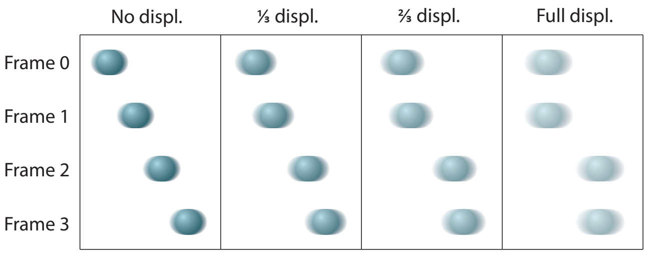


Figure 4: *Four frames sampled using kernels from Fig. 3 for a scene consisting of a ball moving horizontally left to right. Note the unequal spacing between ball positions in the second and third columns, and frame doubling in the fourth column. Since the positions of sampling kernels are displaced but the frames are displayed at equal intervals, odd frames are displayed "too late" with respect to their capture time.*

Given the above definitions, we can define a new interpolated sampling with parameters δ and γ as follows:

$$I_k^{(\delta,\gamma)} = \int_{-\infty}^{\infty} S(t) \cdot \mathrm{rect}_{f,w}^{\gamma} \left(t - \mathcal{T}_f^{\delta}(k) \right) \mathrm{d}t.$$

This interpolation technique enables smooth transition between frame rate $f/2$ and f fps at shutter angle w; however, it is not clear what are the perceptual properties of such an operation. We investigate this issue in Section 5, where we show that our technique approximates in-between frame rates well, and we relate actual frame rate and shutter angle combinations to specific values of δ and γ.

4.4 Arbitrarily Low Frame Rates

The construction described above does not impose any constraints on frame rate f, and in particular the same technique can be applied to an $(f/2)$ Hz display, resulting in interpolation between the rates of $(f/4)$ and $(f/2)$ frames per second. The overlapping kernels of the $(f/2)$-fps emulation (Fig. 2b) can be seen as corresponding to individual frames of a "virtual" $(f/2)$ Hz display, and one can displace them jointly to obtain frame rates between $(f/4)$ and $(f/2)$ fps. This procedure can be repeated indefinitely to obtain arbitrarily low frame rates. In the proposed displacement scheme some frame rates (e.g., 8 fps embedded in 24 fps content) are represented using non-uniformly sampled kernels, even though it would be technically possible to represent them exactly by frame repetition. Although this can potentially lead to sub-optimal results at very low frame rates, where one can easily see individual frames, the benefits of accounting for all such embeddings in the entire frame-rate range are not immediately clear. Thus, for simplicity, we use the "divide-by-two" scheme.

4.5 Preventing Shape Distortion

In our construction, we moved only odd sampling kernels, while keeping even kernels unchanged. This results in a slight positioning error of moving objects along the motion direction, and can cause distortion of the image, particularly visible as slanting of vertical lines. To avoid this effect, in our implementation we displace both kernels symmetrically in opposite directions, which is achieved by modifying function $\mathcal{T}_f^{\delta}$ as follows:

$$\mathcal{T}_f^{\delta}(k) = \begin{cases} t_0 + (k + \delta/2)/f & \text{for even } k, \\ t_0 + (k - \delta/2)/f & \text{for odd } k. \end{cases}$$

4.6 Implementing Arbitrary Sampling Kernels

Although interpolation parameters δ and γ have been defined globally for the whole image, the above equation can be generalized to allow for spatial variation by letting each pixel assume its own δ and γ. This requires that each pixel be sampled at arbitrary time-points with a kernel of arbitrary size. In the case of rendered content, such a sampling could be incorporated directly in the renderer. Modern renderers can efficiently simulate finite-time exposure, and the only additional feature we require is that instead of using a single global temporal sampling kernel, many local sampling kernels are used. However, when only an input video is available, one needs to re-sample it in order to obtain the required sampling kernels. We propose two solutions to this problem: an accurate but costly filtering of a densely-sampled video or an optic-flow-based warping of a regular video.

4.6.1 Dense Input Video

If the temporal resolution of the input video is high (hundreds of frames per second), the re-sampling is straightforward and can be implemented by simple temporal filtering of the input video. Each pixel of each video frame is considered independently, and its value is obtained by averaging pixel values at the corresponding position in all frames that fall within the time interval defined by the kernel. This approach introduces some temporal quantization of the sampling kernel; however, given a sufficiently high input frame rate, this error becomes negligible. The disadvantage of this approach is that generating a densely-sampled video is a costly process.

4.6.2 Sparse Input Video

When sampling a dense input video is not possible, determining the value of a given pixel at an arbitrary time-point is not trivial. In this case, we approximate arbitrary, spatially varying sampling kernels using frame blending followed by optic-flow-based frame warping. The preferred format of the input video for this method is a near-360° shutter, at a relatively high f (e.g., 120 or 96). Such high-frame-rate videos are an emerging standard in the film industry [Brostow and Essa 2001; Richards 2014; Trumbull and Jackson 2013], enabling synthesis of various frame rates and shutter combinations, which is achieved by dropping some of the frames of the original video and blending the remaining ones. For instance, by averaging one, two, three, or four consecutive frames, one obtains the corresponding frame of a 90-, 180-, 270-, or 360-degree, $(f/4)$-fps video, respectively. In-between shutter angles can be approximated by blending between those outputs. The sequences used in the experiment in Sec. 5 were generated assuming such input. Applying this method is also possible for lower-frame-rate videos: for instance, when the input video is a 24-fps, 90-degree one, it can be temporally up-sampled to 96 fps, 360 degree using frame interpolation. Depending on the initial frame rate and shutter angle combination, different kernel sizes can be reproduced with varying degrees of accuracy. At the very least, the input video can be temporally up-sampled ignoring the shutter angle and a simplified version of the below procedure can be implemented, with the first step (frame blending) omitted. The results in Sec. 7 were generated using this approach.

Let V_k denote the k-th frame of the f-fps, 360-degree input video, $K_k \in \mathbb{N}^2 \to \mathbb{R}^+$ and $D_k \in \mathbb{N}^2 \to [0,1]$ the maps of kernel sizes and displacements, respectively, and $F_k, B_k \in \mathbb{N}^2 \to \mathbb{Z}^2$ the corresponding forward and backward optic flow maps (in our experiments we used the technique by Brox et al. [2004] to estimate these). The value at $K_k(i,j)$ is the integration time for frame k and the pixel position (i,j) in seconds multiplied by f (effectively, the number of 360°, f-fps kernels covering the intended integration time), and the value $D_k(i,j)$ is the displacement parameter δ for that pixel. We proceed in two steps. First, we take an input frame corresponding to the desired presentation time, and locally blend it with neighboring frames to approximate the required kernel size (we omit pixel indexing for clarity; all operations are performed pixel-wise):

$$\begin{aligned}\bar{V}_k = (&\mathrm{clamp}(K_k;0,1)\cdot V_k\\ &+\sum_{n=1}^{\infty}\frac{1}{2}\cdot \mathrm{clamp}(K_k-2n+1;0,2)\cdot(V_{k-n}+V_{k+n}))/K_k,\end{aligned}$$

where $\mathrm{clamp}(x;a,b)=\min(\max(a,x),b)$. In the above equation, the preceding and succeeding frames are symmetrically blended with the current frame, resulting in a weighted average between the nearest smaller and the nearest larger integration time possible to achieve by integrating a whole number of frames. For instance, if $K_k = 4.2$, the resulting frame $\bar{V}_k$ equals $(0.6\cdot V_{k-2}+V_{k-1}+V_k+V_{k+1}+0.6\cdot V_{k+2})/4.2$.

Second, we warp the frame by re-projecting each pixel to its position in the past or in the future (depending if the frame is even or odd), with the time-point being determined by the desired kernel displacement at the given pixel:

$$\bar{V}_k(i,j)\mapsto\begin{cases}\hat{V}_k((i,j)+{}^1\!/\!_2\cdot D_k(i,j)\cdot F_k(i,j)) & \text{for even } k,\\ \hat{V}_k((i,j)+{}^1\!/\!_2\cdot D_k(i,j)\cdot B_k(i,j)) & \text{for odd } k.\end{cases}$$

The arrow notation $\bar{V}_k(i,j)\mapsto\hat{V}_k(i',j')$ means, that the pixel in the input image at the position (i,j) is warped to the position (i',j') in the output image. Note that after the warping the actual kernel at any given position in $\hat{V}_k$ is not exactly equal to that given by K_k and D_k for that position: In this approach, the value of V_k at the location (i,j), based on K_k and D_k, is splatted to *another location* (i',j'), whereas in the filtering approach for dense videos, the value $\bar{V}_k(i,j)$ is dependent only on pixel values at the *same location* at different time-points. However, under the assumption that the kernel displacement/size and optical flow are locally constant, the outcome is equivalent in both solutions.

Since this method blends few frames to approximate different kernel sizes, its accuracy in this respect is admittedly lower when compared to the dense video approach. However, it has the advantage of a relatively low computation cost, enabling a real-time implementation, e.g., in TV sets or computer games. Our method relies on optical flow estimation; therefore, artifacts typical for such algorithms can at times leak to our results. Since tuning of the flow algorithm was not the focus of our work, we used a non-optimized optical flow algorithm with a constant set of parameters and no flow reliability estimation. In commercial applications either some manual input is assumed (film post-production) or fine-tuned algorithms with low artifact visibility are used (TV sets), which likely include smart strategies of determining when such an upsampling may fail, and reverting to the original content in such cases. Note that many TV sets enable the frame interpolation mode by default, and artifacts are rarely spotted by non-professional users.

5 Parameter Estimation

To investigate the perceptual effect of our interpolation technique, we establish a mapping between combinations of actual frame rates and shutter angles and the interpolation parameters δ and γ in the range 24–96 fps. Although our technique is not limited to $f = 96$ (e.g., $f = 48, 60, 100,$ or 120 could be used as well), we believe that this is the most interesting scenario for our method, because it allows for an exact emulation of both standard 24 fps and HFR 48 fps. We derived the mapping in the following calibration experiment.

Subjects and Experimental Setup Ten subjects, including two authors, took part in the experiment. They were all members of the computer graphics group. We used an Asus PG278Q display (27-inch diagonal, native resolution 2560 × 1440 px, maximum refresh rate 144 Hz) and an Nvidia GeForce GTX 970 graphics card. This configuration supports Nvidia G-Sync technology, which enables the system to refresh the display as soon as the frame has been rendered, without waiting for the next refresh cycle of the display. Thus, by putting the process to sleep for an appropriate number of milliseconds, we were able to set the display programmatically to any frame rate below 144 Hz on the fly. The subjects were seated ca. 50 cm from the display, but were allowed to freely change their position. The experiment was conducted in controlled office lighting conditions.

Stimuli and Procedure The stimulus was a vertical $100 \times 1440\,\mathrm{px}$ light-gray bar moving left-to-right on a dark-gray background. When the bar reached the right end of the display, the motion was restarted from the left end of the display. The subjects could alternate between the reference bar and the test bar by pressing the left and the right arrow key, respectively. Both bars were moving with velocity $v \in \{256\,\mathrm{px/s},\, 512\,\mathrm{px/s},\, 1024\,\mathrm{px/s}\}$. The reference bar was displayed with veridical frame rate $f_\mathrm{r} \in \{29, 34, 40, 68\}$ and normalized shutter angle $w_\mathrm{r} \in \{0.25, 0.5, 0.75\}$. The test bar was always displayed using our technique at frame rate $f_\mathrm{t} = 96\,\mathrm{fps}$. Kernel displacement of the test bar could be adjusted via parameter $d \in [1, 4]$ by pressing the plus and the minus key, and shutter angle w_t could be adjusted in the range of $[0, 4]$ by pressing the '[' and ']' keys. Values of $d \in [1, 2]$ corresponded to $\delta \in [0, 1]$, whereas values of $d \in [2, 4]$ corresponded to $\delta \in [0, 1]$ assuming a "virtual" frame rate of $f/2 = 48\,\mathrm{fps}$ achieved by joint displacement of overlapping kernels (see Sec. 4.4 for explanation). In a single trial, the participant was asked to adjust the kernel displacement d and shutter angle w_t of the test bar so that its appearance matched the appearance of the reference bar as closely as possible, and confirm the settings with the 'Enter' key. The whole session consisted of all $3 \cdot 4 \cdot 3 = 36$ possible trials in random order, and the time to perform the task was not limited. We did not test $f_\mathrm{r} \in \{24, 48, 96\}$, since our method can emulate these rates exactly.

Results and Discussion The results of the calibration experiment are presented in Fig. 5. As can be seen, d is approximately inversely proportional to the reference frame rate; however, for 34 and 40 fps this value tends to be lower. This is accompanied by significantly increased blur in comparison to what would be predicted by simple matching of the absolute exposure time. In our experience, the most important factor determining the similarity of the two bars for frequencies between 24 and 48 fps was the perceived intensity of judder at the bar edges. As shown in Fig. 6, the displacement values at the black solid line in Fig. 5 result in the same juddering area. However, the judder of our emulation has lower frequency than that of the reference stimulus (24 Hz vs. 29, 34, or 40 Hz). We hypothesize that these deviations of judder area combined with increased blur compensate for this difference in frequency and result in judder of equal perceived strength.

When the frame rate of the stimulus exceeds the critical flicker frequency [Kalloniatis and Luu 2009], the changing signal is averaged by the visual system, and the bar appears blurred (so-called hold-type blur). Thus, for the highest frame rate (68 fps), the dominant parameter is the amount of blurring at the edges, since virtually no judder is visible in this case. A similar effect was observed by Navarro et al. [2011].

The obtained data points can be interpolated and used to define improved correspondence between intended frame rate and interpolation parameters δ and γ. Although the experiment involved a modest number of subjects, the larger evaluation study in Sec. 6 proves that the scale of this calibration procedure was sufficient for our purposes.

6 Evaluation

In the previous section, we found the optimal mapping between frame rates and the values of the kernel displacement parameter for synthetic stimuli. However, it still remains to be shown, that our frame rate emulation leads to possibly similar appearance for real-world content. In this section we present a perceptual evaluation experiment in which we compare the proposed technique against a baseline method. Thirty-five naïve, non-expert, paid subjects took part in the experiment. All had normal or corrected-to-normal vision. The experimental setup was the same as in the calibration experiment.

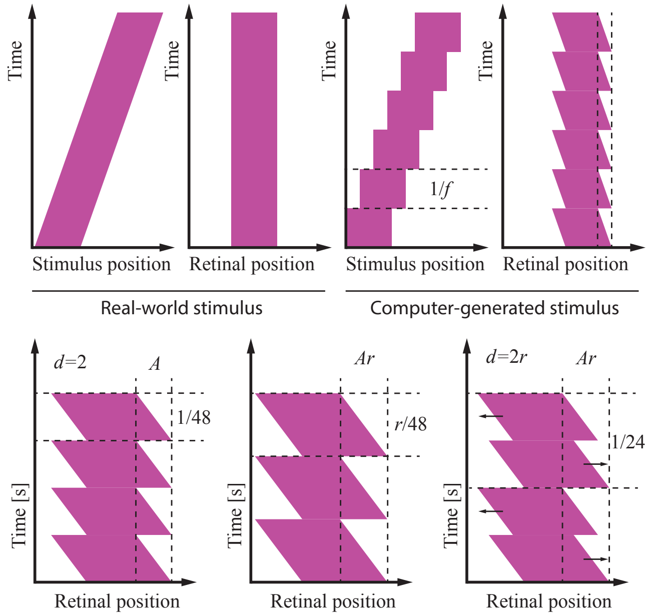


Figure 6: *Top: Comparison of a real-world stimulus (left) and a computer-generated stimulus (right). In each pair we show the horizontal position of a moving vertical bar. Due to smooth pursuit eye motion, the stimulus' image is stabilized on the retina. While real-world stimuli generate constant signal on the retina, computer-generated stimuli have regions of time-varying periodic signal near the edges, because the bar "stays behind" due to its position changing in discrete steps. One such region is delineated by the vertical dashed lines. Depending on the frame rate of the display, this will cause judder and/or hold-type blur. Bottom: at $d = 2$ we achieve an exact emulation of 48 fps, which has a certain juddering area of width A (left). In the middle figure, some lower frame rate (48/r) fps yields a juddering area of width Ar. Setting the displacement parameter d in our emulation to 2r (right), which corresponds to a position on the black solid line in Fig. 5, gives a juddering area of equal width; however, the frequency of flicker is lower (24 Hz).*

Stimuli and Procedure Eight real-world, 96-fps sequences chosen to cover a wide range of different objects and types of motion were used as stimuli (see Fig. 7 and the supplemental material). All were captured with a near-360-degree shutter angle, except two (*Terrace* and *Chairs*) for which the shutter angle was less than 360 degrees due to camera limitations, but those sequences were treated in the same way as the remaining six sequences. We used the OpenCV 2.4.8 implementation of Brox et al.'s [2004] algorithm ($alpha = .2$, $gamma = 50$, $scale = .8$, $inner = 10$, $outer = 77$, $solver = 10$), to estimate optic flow in the sequences.

The reference sequences (obtained via optic-flow-based frame interpolation of the original video) were rendered using frame rate $f_r \in \{29, 34, 40, 68\}$ and shutter $w_\mathrm{r} \in \{f_\mathrm{r}/96, 2 \cdot f_\mathrm{r}/96\}$ (except for $f_\mathrm{r} = 68$, where only $w_\mathrm{r} = 68/96$ was used). The test sequences were synthesized using our technique (see Sec. 4.6.2) at frame rate $f_\mathrm{t} = 96\,\mathrm{fps}$, with displacement d and shutter w_t locally adjusted according to the velocities in the video, as determined in the calibration

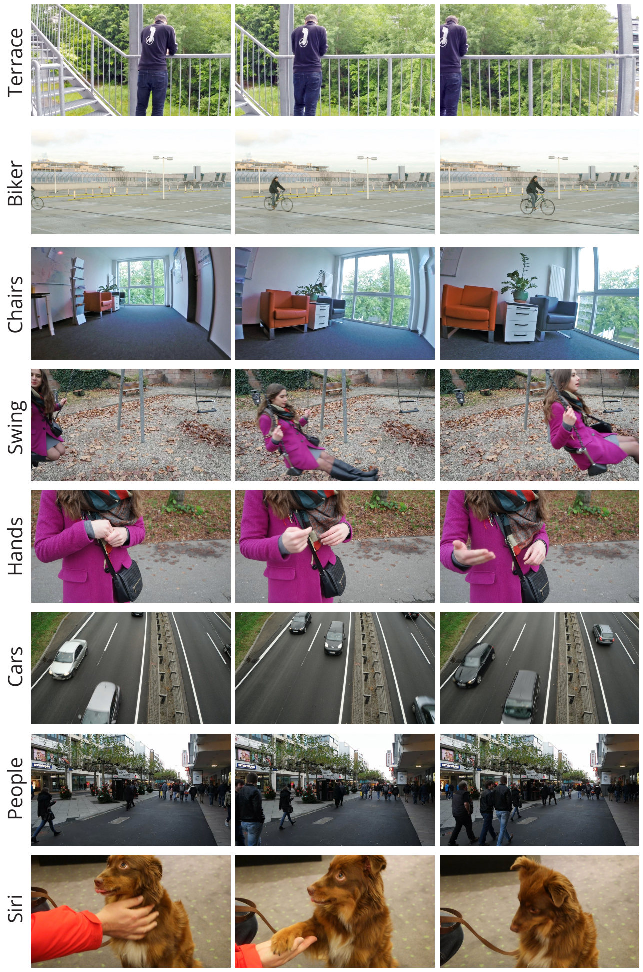


Figure 7: *Scenes used in the evaluation experiment. The first group of subjects saw* Terrace, Biker, *and* Chairs *scenes, while the other group of subjects saw the remaining five scenes.* Biker *sequence:* © *Stefan Grandinetti and Harald Brendel,* Swing *sequence:* © *Krzysztof Templin.*

experiment (Fig. 5). The comparison baseline sequence was rendered using frame rate $f_b \in \{48, 96\}$ when $f_r = 68$ and $f_b \in \{24, 48\}$ otherwise. The value of the baseline shutter w_b was set to match the absolute exposure time of the reference video (the same amount of blur).

The subjects could switch between the reference, test, and the comparison sequence using the arrow keys, with the 'Up' key corresponding to the reference sequence, and the 'Left'/'Right' keys corresponding to the test and comparison sequence in random arrangement. In a single trial, the subject was asked to select one of the two sequences that *looked more similar* to the reference sequence and confirm the choice with the 'Enter' key. The videos were divided into two groups, one consisting of three videos (*Biker*, *Terrace*, and *Chairs*) and one consisting of five videos (the remaning scenes), and each subject was assigned to one of the groups. One session consisted of all possible trials in random order ($3 \cdot 7 \cdot 2 = 42$ in the first group and $5 \cdot 7 \cdot 2 = 70$ in the second group).

Before the experiment, a control session was performed in which the frame rate of the reference and the test sequence was set to either 24, 48, or 96 fps and the comparison sequence was set to one of the remaining two frame rates (thus the test sequence was identical to the reference, while the comparison sequence had a significantly different frame rate). Three of the subjects were unable to perform above the chance level in this setting and were subsequently excluded from our analysis. In the end, each video in each group was judged by sixteen subjects.

Results and Discussion The results of the experiment are presented in Figure 8. In the analysis of the results we included only comparisons with this baseline frame rate that on average performed better for the given reference frame rate. For instance, 29 fps reference sequences were on average much better approximated by the corresponding 24 fps sequence than by the 48 fps one; thus, we excluded 29-48 fps comparisons from the analysis. In general, for 29, 34, 40, and 68 fps conditions we assumed baselines of 24, 48, 48, and 100 fps, respectively. We coded the subjects' responses as 0-1 binary scores, 1 meaning that in the given trial the subject preferred our solution over the baseline solution. By averaging scores first within subjects and then between subjects, we obtained an average score of 0.74 in the first group and 0.63 in the second group. Both results are significant: a one-sided sign test ($n = 16$) rejects the null hypothesis of a distribution with a 0.5 median at $p < 0.0003$ in the first group and $p < 0.003$ in the second group. We also calculated scores for each scene separately. As expected, the scenes with a steady, easy-to-track motion (e.g., *Terrace*, *Swing*) were scored higher than those with complicated motion (e.g., *Siri*, *People*). In particular, *Terrace* sequence contains a steady, whole-image motion due to camera panning. Such motion, however, is very common in cinematography – camera pans, tilts, zooms etc. constitute around 20% of all shots [Salt 2009, p. 371]. Nonetheless, the good score of *Hands* sequence shows that scenes with non-trivial motion can also benefit from using our method.

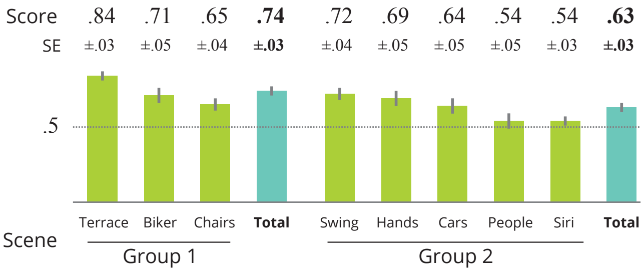


Figure 8: *Results of the evaluation experiment. The scores in the first row were calculated by averaging 0-1 scores within subjects and then between subjects. The second row shows standard errors for between-subject averages ($n = 16$). The scores were calculated for each scene separately (green bars) and for all scenes within the group (teal bars). The error bars in the plot are two standard errors wide.*

The baseline methods used nearest standard cinematic frame rates and had matching amounts of blur, which can be considered the state of the art in terms of matching the film look [Tessive 2014]. An alternative baseline solution would be to play the motion-interpolated reference video at 24, 48, or 96 frames per second by dropping or repeating frames. This method, however, introduces temporal and spatial artifacts (as discussed in Sec. 4.2 and shown in the supplemental video); thus, we did not consider it to be a feasible solution to the problem addressed in this paper.

In general, our technique turned out to be more similar to the reference than the baseline sequences. While our parameter calibration in Sec. 5 could be extended by taking into account more parameters, it covered the most relevant part of the complex images in the context of frame rate perception – high-contrast edges. Since our technique already provides a very good emulation of frame rate looks for natural (live-action) videos, we think that investigating different spatial frequencies, contrasts, and motion types would unnecessarily complicate the method.

We used the warping approach to generate sequences in this experiment; thus, the optical flow estimation artifacts affected the visual quality of the resulting sequences. Although we used the same optical flow estimates to generate all three sequences (i.e., reference, comparison, and test) in each trial, the slight differences in the artifact visibility might have influenced the result of the experiment.

7 Advanced Applications

Besides the whole-image frame-rate adjustment (sequences used in the evaluation in Sec. 6), which provides filmmakers with a fine-grained control of the overall film appearance (motion smoothness vs. cinematic look trade-off), we propose two advanced applications of our frame rate interpolation technique. First, it can be used by the artist to locally apply manual tweaks to the video, based on his or her artistic vision (see Fig. 9 and sequences *Bridge*, *Door*, and *Rope* in supplemental material). In contrast to standard techniques, where the artist is forced to choose from a limited set of possible frame rates and the changes between frame rates have to be discontinuous, our technique enables arbitrary frame rate variation. Similarly, smooth temporal variation of the frame rate can help make the moment of transition unnoticeable when an abrupt frame-rate change is not desired.

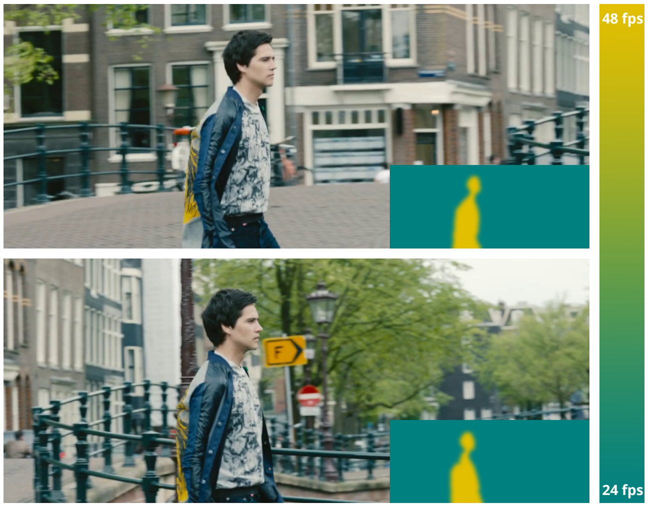


Figure 9: *The frame rate in the video can be varied according to manual tweaks provided by the artist.* Bridge *sequence: (CC) Blender Foundation | mango.blender.org*

In the second application, the velocities in the sequence can be automatically analyzed and the appropriate frame rate can be applied locally. For instance, depending on the camera parameters such as focal length and frame rate, there are certain recommendations as to the maximum comfortable on-screen speed of any object in the scene [Hummel 2002, p. 887] [Samuelson 2014, p. 314]. Using these guidelines, our technique can automatically minimize the frame rates across the screen in order to maximize the cinematic look, yet without introducing objectionable artifacts (see Fig. 10 and sequences *Deer*, *Thom*, and *Tucson* in supplemental material).

In all examples in this section, we emulate frame rates between 24 fps and 48 fps. Since the input videos were shot at 24 fps with an unknown shutter angle, we decided to preserve the original blur and we assumed a displacement parameter inversely proportional to the desired frame rate. In the automatic adjustment examples, we computed a map of flow magnitudes in the input and blurred them using a Gaussian blur (the sigma parameter equaled 40 pixels). The values between 0 and 30 px/frame were linearly transformed to the range 24–48 and used as a spatio-temporally varying frame rate map guiding our method (false-color visualization in the video). As a result, the cinematic look is largely preserved, while the regions with most flicker are de-emphasized.

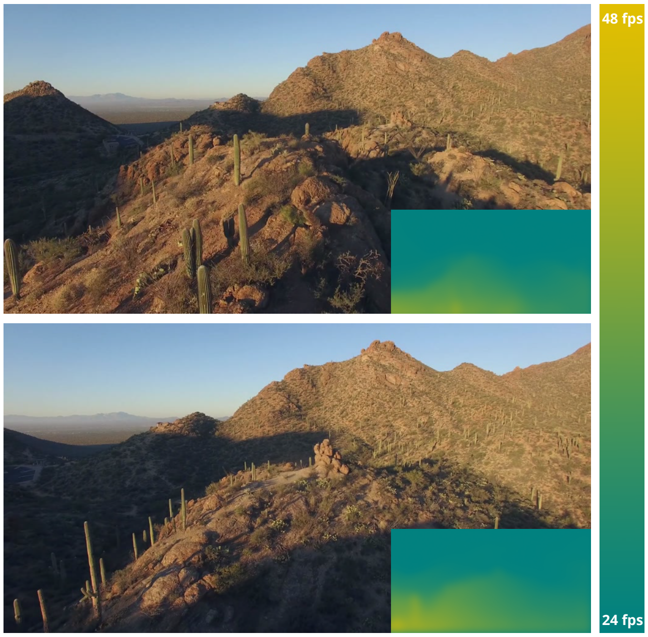


Figure 10: *The velocities in the video can be analyzed automatically and mapped linearly to frame rates in order to reduce judder while preserving the cinematic look.* Tucson *sequence: (CC) Roger Williams*

8 Conclusions and Future Work

In this paper we introduced a technique for emulating the whole range of frame rates on a single-frame-rate display. It allows for smooth spatio-temporal variation of the frame rate for maximum creative freedom of filmmakers. To our knowledge this is the first technique of its kind, and we believe that such a fine control will be crucial for full leveraging of high-frame-rate cameras. Furthermore, we demonstrated how our technique can automatically minimize the frame rate based on the local velocity distribution, ensuring a maximal “cinematic look” without introducing objectionable artifacts. The technique is easy to implement and can be used both for animation and live-action films. Since usually the frame rate of the input video will not be high enough to apply a straightforward filtering (Sec. 4.6.1), the optic-flow-based image warping needs to be used (Sec. 4.6.2). In this case the technique inherits limitations of optic flow estimation algorithms: large velocities, non-Lambertian surfaces, or complex lighting can all negatively affect the results. Nevertheless, the widespread use of semiautomatic (film production) or fully automatic (TV sets) optic-flow-based techniques proves that usually this is not a problem. Our technique can be also easily integrated in TV sets, which already apply frame interpolation, to adjust the appearance of the videos depending on the content characteristics or user preference. Naturally, higher input frame rates improve the accuracy of the flow estimation and shutter synthesis, but even a 24-fps video can be used as the input when precise control over the shutter angle is not required.

The key component of our solution is the temporally varying displacement introduced to the sampling filter. It allows us to control the perceived strength of judder, which is one of the distinctive features of lower-frame-rate motion pictures. As the target frame rate goes beyond 60 fps the role of this effects gets smaller, and thus, the contribution of the kernel displacement technique is limited. On the other hand, there is evidence that humans perceive flicker at rates as high as 500 Hz [Davis et al. 2015]. This opens up an interesting question: what is the optimal method of frame rate reduction at very high frequencies?

As future work we intend to investigate the utility of our technique for stereoscopic presentation. Although it seems to be immediately feasible, the image separation protocols between eyes, for example in time-sequential shutter glasses, might cause additional motion perception artifacts that would need to be taken into consideration [Hoffman et al. 2011]. We calibrated our technique using a standard-sized desktop display. Although the angular coverage in our setup was relatively large (60 degrees), it would be worthwhile to evaluate the performance of our technique in the theatrical environment as well. An interesting avenue for future research would be to explore the possibilities of integration of our algorithm with specialized "motion-aware" cameras [Gupta et al. 2010; Agrawal et al. 2010; Taguchi et al. 2012; Tambe et al. 2013]. Finally, it is unclear how the human perception of different frame rates changes over time – it is likely, that some form of frame-rate adaptation exists and prior experience might play a role as well. In the future it would be interesting to investigate the long-term temporal properties of the frame-rate perception and to extend the presented technique by taking into account any such effects.

Acknowledgments

We would like to thank Prof. Stefan Grandinetti from Hochschule der Medien (Stuttgart) and Harald Brendel from ARRI Cine Technik (Munich) for providing us with the *Biker* video sequence.

References

AGRAWAL, A., VEERARAGHAVAN, A., AND RASKAR, R. 2010. Reinterpretable imager: Towards variable post-capture space, angle and time resolution in photography. *Computer Graphics Forum 29*, 2, 763–772.

BROSTOW, G. J., AND ESSA, I. 2001. Image-based motion blur for stop motion animation. In *Proc ACM SIGGRAPH*, 561–566.

BROWNLOW, K. 1980. Silent films: What was the right speed? *Sight and Sound*, Summer, 164–167.

BROX, T., BRUHN, A., PAPENBERG, N., AND WEICKERT, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, vol. 3024 of *Lecture Notes in Computer Science*, 25–36.

CHAI, J.-X., TONG, X., CHAN, S.-C., AND SHUM, H.-Y. 2000. Plenoptic sampling. In *Proc ACM SIGGRAPH*, 307–318.

DALY, S., XU, N., CRENSHAW, J., AND ZUNJARRAO, V. J. 2014. A psychophysical study exploring judder using fundamental signals and complex imagery. In *SMPTE Annual Technical Conference*.

DALY, S. 1998. Engineering observations from spatiovelocity and spatiotemporal visual models. In *Human Vision and Electronic Imaging III*, SPIE Vol. 3299, 180–191.

DAVIS, J., HSIEH, Y.-H., AND LEE, H.-C. 2015. Humans perceive flicker artifacts at 500 Hz. *Scientific Reports 5*.

DE LANGE, H. 1958. Research into the dynamic nature of the human fovea - Cortex systems with intermittent and modulated light. I. Attenuation characteristics with white and colored light. *J. Opt. Soc. Am. 48*, 11, 777–783.

DIDYK, P., EISEMANN, E., RITSCHEL, T., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2010. Perceptually-motivated real-time temporal upsampling of 3D content for high-refresh-rate displays. *Computer Graphics Forum (Proc. Eurographics 2010) 29*, 2, 713–722.

DISNEY RESEARCH, 2015. *Lucid Dreams of Gabriel*. http://www.disneyresearch.com/luciddreamsofgabriel/flowTime.html.

FENG, X.-F. 2006. LCD motion blur analysis, perception, and reduction using synchronized backlight flashing. In *Human Vision and Electronic Imaging XI*, SPIE Vol. 6057, M1–14.

FUCHS, M., CHEN, T., WANG, O., RASKAR, R., SEIDEL, H.-P., AND LENSCH, H. P. 2010. Real-time temporal shaping of high-speed video streams. *Computers & Graphics 34*, 575–584.

GUPTA, M., AGRAWAL, A., VEERARAGHAVAN, A., AND NARASIMHAN, S. G. 2010. Flexible voxels for motion-aware videography. In *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6311 of *Lecture Notes in Computer Science*. 100–114.

HOFFMAN, D. M., KARASEV, V. I., AND BANKS, M. S. 2011. Temporal presentation protocols in stereoscopic displays: Flicker visibility, perceived motion, and perceived depth. *J Soc Inf Disp 19*, 3, 271–297.

HUMMEL, R., Ed. 2002. *American Cinematographer Manual (8th Edition)*. ASC Holding Corp.

KALLONIATIS, M., AND LUU, C. 2009. Temporal resolution. *http://webvision.med.utah.edu/temporal.html*.

MÄKELÄ, P., ROVAMO, J., AND WHITAKER, D. 1994. Effects of luminance and external temporal noise on flicker sensitivity as a function of stimulus size at various eccentricities. *Vision Research 34*, 15, 1981–91.

NAVARRO, F., CASTILLO, S., SERÓN, F. J., AND GUTIERREZ, D. 2011. Perceptual considerations for motion blur rendering. *ACM Trans. Appl. Percept. 8*, 3, 20:1–20:15.

QUESNEL, D., LANTIN, M., GOLDMAN, A., AND ARDEN, S., 2013. An exploration into the creation of variable frame rate (VFR) stereoscopic 3D narrative productions. Emily Carr University of Art and Design. http://research.ecuad.ca/s3dcentre/projects/hfr-high-frame-rate-research/, accessed Apr 27, 2015.

RASKAR, R., AGRAWAL, A., AND TUMBLIN, J. 2006. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Trans. Graph. 25*, 3, 795–804.

RICHARDS, D. 2014. 120 fps as a universal production format for motion pictures. In *SMPTE Annual Conference Abstracts*. http://edas.info/p17109, accessed Jan 6, 2015.

SALT, B. 2009. *Film Style and Technology: History and Analysis*. Starword.

SAMUELSON, D. 2014. The mathematics of cinematography. In *Hands-on Manual for Cinematographers*. CRC Press.

STENGEL, M., BAUSZAT, P., EISEMANN, M., EISEMANN, E., AND MAGNOR, M. 2015. Temporal video filtering and exposure control for perceptual motion blur. *Visualization and Computer Graphics, IEEE Transactions on 21*, 2, 1–11.

STICH, T., LINZ, C., WALLRAVEN, C., CUNNINGHAM, D., AND MAGNOR, M. 2008. Perception-motivated interpolation of image sequences. In *Proc. APGV*, 97–106.

STUMP, D. 2014. Frame rates and aspect ratios. In *Digital Cinematography: Fundamentals, Tools, Techniques, and Workflows*. Focal Press, Burlington, MA.

SUNG, K., PEARCE, A., AND WANG, C. 2002. Spatial-temporal antialiasing. *IEEE Transactions on Visualization and Computer Graphics 8*, 2, 144–153.

TAGUCHI, Y., AGRAWAL, A., AND TUZEL, O. 2012. Motion-aware structured light using spatio-temporal decodable patterns. In *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., vol. 7576 of *Lecture Notes in Computer Science*. 832–845.

TAMBE, S., VEERARAGHAVAN, A., AND AGRAWAL, A. 2013. Towards motion aware light field video for dynamic scenes. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 1009–1016.

TELLEEN, J., SULLIVAN, A., YEE, J., WANG, O., GUNAWARDANE, P., COLLINS, I., AND DAVIS, J. 2007. Synthetic shutter speed imaging. *Comput. Graph. Forum 26*, 3, 591–598.

TESSIVE, 2014. Time Filter, Time Shaper. http://tessive.com.

TRUMBULL, D. H., AND JACKSON, B., 2013. Method and apparatus for photographing and projecting moving images. US Patent US8363117 B2.

VANGORP, P., CHAURASIA, G., LAFFONT, P.-Y., FLEMING, R., AND DRETTAKIS, G. 2011. Perception of visual artifacts in image-based rendering of façades. *Computer Graphics Forum (Proc. of the EGSR) 30*, 4, 1241–1250.

WATSON, A. B. 2013. High frame rates and human vision: A view through the window of visibility. *SMPTE Motion Imaging Journal 122*, 2, 18–32.

WESTERINK, J. H., AND TEUNISSEN, K. 1995. Perceived sharpness in complex moving images. *Displays 16*, 2, 89 – 97.

WILCOX, L. M., ALLISON, R. S., HELLIKER, J., DUNK, B., AND ANTHONY, R. C. 2015. Evidence that viewers prefer higher frame-rate film. *ACM Trans. Appl. Percept. 12*, 4, 15:1–12.

ZWICKER, M., MATUSIK, W., DURAND, F., PFISTER, H., AND FORLINES, C. 2006. Antialiasing for automultiscopic 3D displays. In *Proc. EGSR*, 73–82.