Abstract—Media technology, in particular video recording and playback, keeps improving to provide users with high-quality real and virtual visual content. In recent years, increasing the temporal sampling rate of videos and the refresh rate of displays has become one focus of technical innovation. This raises the question, how high the sampling and refresh rates should be? To answer this question, we determine the minimum temporal sampling rate at which a video should be presented to make temporal sampling imperceptible to viewers.

Through a psychophysical study, we find that this minimum sampling rate depends on both the speed of the objects in the image plane, and the exposure time of the recording camera. We propose a model to compute the required minimum sampling rate based on these two parameters. In addition, state-of-the-art video codecs employ motion vectors from which the local object movement speed can be inferred. Therefore, we present a procedure to compute the minimum sampling rate given an encoded video and camera exposure time.

Since the object motion speed in a video may vary, the corresponding minimum frame rate is also varying. This is why the results of this paper are particularly applicable when used together with adaptive frame rate computer generated graphics or novel video communication solutions that drop insignificant frames. In our experiments, we show that videos played back at the minimum adaptive frame rate achieve an average bit rate reduction of 26% compared to constant frame rate playback, while perceptually no difference can be observed.

Index Terms—Video Sampling Frequency, Adaptive Sampling, Perception Limits.

I. INTRODUCTION

Industry and science strive to make technology underlying visual media consumption and interaction imperceptible to the user. To make this happen, latencies need to be reduced, and spatial and temporal resolution need to be increased. Latencies are particularly relevant in interactive media consumption such as vision-based remote control, video conferencing, cloud gaming, or content consumption using a head-mounted display (HMD). State-of-the-art systems offer significant potential for improvement in this aspect [1]. There are novel systems that reduce latency by recording video sequences at high frame rate, and dropping insignificant frames (adapting the temporal sampling rate) even before encoding [2]. This frame dropping process decouples the display frame rate from the recording frame rate, and consequently from exposure time. One important parameter for driving the frame dropping process is the required minimum display frame rate\(^1\). In this paper we study the impact of the camera exposure time and the object motion velocity in the image plane on the required minimum display rate. We have not found any studies examining their combined effect on the minimum display rate.

More specifically, we answer the question of the required minimal displaying frame rate for a given constant camera exposure time (and constant camera frame rate) and object motion velocity. To this end, the paper analyses temporal sampling in video recording and display as well as how camera exposure time and scene motion affect the frame rate required to give the user the impression of continuous motion. These insights allow us to improve video compression efficiency by skipping perceptually insignificant video frames from a camera that is recording at a constant frame rate. Effectively, the algorithm adapts the video frame rate to scene motion and camera exposure time. The proposed adaptive frame skipping is designed to have no influence on what the video consumer perceives compared to a video played back at the original constant camera frame rate.

A. Related Work

Researchers in the field of psychophysics have thoroughly investigated how the human visual system creates the impression of an object in motion. However, the community has not yet agreed on the actual neural processes. This section summarizes research related to human motion perception relevant for this paper.

Johansson [3] discusses a major difference between the human eye and camera systems: the human eye does not have a shutter, and accordingly has unlimited exposure time. Nevertheless, humans perceive their moving environment not blurry, but perfectly clear. The author states that neural algorithms which process the continuous exposure data from the photoreceptors on the eye’s retina are responsible for this impression.

Derrington et al. [4] provide a review of the state-of-the-art in motion perception research. The authors find evidence only

\(^1\)At the minimum frame rate, the user should perceive perfectly continuous (fluid/smooth) motion, and effects from temporal sampling (jitter/jerkiness) should be imperceptible.
to support the assumption of a first-order motion detector on the retina, and propose that there is no evidence for processing steps that are more complex. First order motion detection means determining motion based on the spatiotemporal correlation of brightness or color changes. They review three candidate models for this detection. The paper does not contain a discussion of perception limits.

The results of Derrington et al. [4] are in contrast to the findings of Ledgeway and Smith [5], who found evidence to support that there are different pathways for processing first-order and second-order motion. Second-order motion is based on spatiotemporal variations such as depth, contrast and relative motion which do not yield a systematic motion in the Fourier domain. It is thus also named "texture-based" motion perception. Examples of such signals are given by Chubb and Sperling [6].

Seiffert and Cavanagh [7] conducted experiments to separately analyze first-order as well as second-order motion. They found that for first-order motion detection, the velocity of an object, and not the spatial displacement determines a lower bound for the perception of motion. Second-order motion perception is performed by tracking object features over time, and its lower thresholds, in contrast to first-order perception, are not affected by the speed of the object to be detected, but by a minimal position change.

Adelson and Bergen [8] suggest energy models for modeling motion perception. The authors conceptualize two dimensional motion as three dimensional patterns in the x-y-t space. Therefore, motion perception is analogous to identifying an orientation in the x-y-t space. They apply quadrature pair filters to extract spatiotemporal energy which is then further processed to determine perceived motion.

The paper by Watson et al. [9] is the most relevant reference for this work, it is hence discussed in greater detail. First, Watson et al. [9] define the Window of Visibility (WoV), depicted as gray rectangle in Fig. 1. The window shows the perception thresholds in one spatial frequency domain (\(\omega\) [cycles/degree]) and the temporal frequency domain (\(f\) [Hz]). Visual information within the window is more or less visible to the user, everything outside is not. Thus, the WoV can be seen as a low-pass filter. The shape of the window indicates that the perception thresholds of temporal and spatial frequency are independent [9]. For simplicity, the WoV covers only one spatial dimension, but can be extended in a straightforward manner to two spatial dimensions. Imagine a second spatial dimension perpendicular to the paper plane. In the plane spanned by the two spatial dimensions, the shape of the WoV is a disk because (as we show in Section III-D) only motion magnitude, not motion direction is relevant for the perception limit. Together with the rectangular WoVs in both spatiotemporal planes, the disk in the spatial plane yields the **Cylinder of Visibility** in the three dimensional (two spatial dimensions, one temporal dimension) space.

In [9], the human temporal flicker frequency threshold \(f_{thr}\) was found to be \(f_{thr,1} = 30\) Hz and \(f_{thr,2} = 33\) Hz for two observers. Stationary temporal contrast fluctuations with a frequency higher than \(f_{thr}\) can not be seen. Analogously, \(\omega_{thr}\) represents the human spatial sensitivity threshold, where the cycles per degree refer to the point of view of the human. In their experiments, Watson et al. [9] find that the thresholds are at \(\omega_{thr,1} = 6\) [cycles/degree] and \(\omega_{thr,2} = 13\) [cycles/degree] for two observers. The authors presume that the low contrast of the display causes these rather low threshold values. Experiments with more test subjects should lead to more generally valid numbers, the preceding values are only examples.

One-dimensional motion of a point with constant velocity \(v\) [degree/second] yields a straight line with slope \(-1/v\) in the spatiotemporal frequency domain [9], see the diagonal line passing through the origin in Fig. 1a. This model assumes the point to exhibit infinitely high spatial frequencies. Time discretization or temporal sampling of the spatiotemporal process leads to periodic replication of the original signal (see additional lines in Fig. 1a), where the replication period in frequency domain is the sampling frequency \(f_s\). In Fig. 1a, we can see that the periodic replications intersect with the WoV, so they pass through the pass-band of the low-pass filter. These aliasing artifacts are what humans perceive as jitter. Therefore, if the frequency spectra of the replications lie outside the WoV, humans will not be able to distinguish the time-sampled representation of the moving object from the continuous representation. The authors in [9] derived a minimum temporal sampling frequency, named critical sampling frequency \(f_c\), which alleviates aliasing to a level below human perception thresholds. In the following we briefly review the findings from [9].

There will be imperceptible aliasing if the replicated spectra just touch the corners of the WoV, as illustrated in Fig. 1b. To satisfy this, the temporal sampling frequency \(f_{s,1}\) has to be equal to or greater than the critical sampling frequency

\[
f_{s,1} \geq f_c = f_{thr} + v \cdot \omega_{thr}. \tag{1}
\]

as can be derived from Fig. 1b. As a result of the point symmetry of the WoV and the spectrum of the moving object, this condition also satisfies the first periodic replication at \(-f_s\). If the maximum spatial frequency \(\omega_0\) of the object is lower than the spatial perception threshold \(\omega_{thr}\) (see Fig. 1b), the condition for the sampling frequency \(f_{s,2}\) accordingly changes to

\[
f_{s,2} \geq f_c = f_{thr} + v \cdot \omega_0. \tag{2}
\]

Merging conditions (1) and (2) yields the general condition

\[
f_s \geq f_c = f_{thr} + v \cdot \min(\omega_{thr}, \omega_0) \tag{3}
\]
as the lower bound for the temporal sampling frequency. The consequential next step is to limit the temporal frequency to, e.g., \(f_0\). This is done in [9], but does not give further insight in our context.

**B. Building on Watson’s Work**

Eq. (3) leaves us with two key takeaways: first, objects with high speed \(v\), yielding a flatter line in Fig. 1, require higher temporal sampling frequencies \(f_s\) to attenuate aliasing artifacts, making jitter imperceptible. Second, at a given speed \(v\) and all spatial frequencies below the perception threshold \(\omega_{thr}\), objects with high maximum spatial frequencies (sharp edges)
where $f_s$ is chosen large enough (thin lines), or due to the limited spatial frequency $\omega_0$ of the signal (thick lines).

Fig. 1: Window of visibility [9] depicted as gray rectangle. The diagonal line (slope $-1/v$) passing through the origin represents a point moving at constant velocity $v$. Due to temporal sampling, the diagonal line is replicated at integer multiples of the sampling frequency $f_s$.

require higher temporal sampling frequencies than objects with lower maximum spatial frequencies (blurry/smooth edges).

Watson et al. [9] did not investigate the influence of the exposure time of a camera. However, camera exposure time influences the maximum spatial frequency, as described in the following. For a better understanding, let us first outline how a (video) camera records an image. In the classical camera model, a shutter prevents light from falling onto the photosensitive material at all times except for the exposure time, also called light integration period. During this time, typically a fraction of a second, the shutter is open and photons keep falling onto the photosensitive material.

In modern CMOS/CCD implementations, the photosensitive material is replaced by light sensors that record images in two-dimensional, usually rectangular pixel arrays. Each pixel can electronically be set to integrate light (shutter open, exposure), or to ignore incoming photons (shutter closed). This replaces the mechanical shutter.

If during the exposure time, the rate at which photons fall onto a pixel is constant, the output from this pixel will perfectly represent what can actually be seen from the pixel’s perspective. If the rate of photons changes during exposure, for example if the camera or object is moving, the pixel’s output will represent the average light intensity recorded during the exposure period. This is why quickly moving objects can appear blurred in photos and videos, exhibiting a limited maximum spatial frequency.

The work by Watson et al. [9] provides the theoretical basis for this paper, but it does not answer how limits such as the maximum occurring spatial frequency $\omega_0$ can be computed with given equipment parameters, or if they exist at all. In real-world applications, imperfect filters, spatial discretization and noise occur. In particular, an object moving through the field of view of a camera with finite exposure time will become blurred, its spatial patterns low-pass filtered. However, as we show in the following, blurring caused by exposure time integration will not have an upper bound in terms of spatial frequency, if the original signal’s spatial frequency was not limited.

To illustrate this, imagine the one-dimensional spatial signal $g(x) = u(x - x_0)$ depicted in the top left graph in Fig. 2. The following explanations extend naturally to more spatial dimensions. Function $u(\cdot)$ denotes the Heaviside step function or unit step function, which is equal to zero for negative arguments, and equal to one for nonnegative arguments. For simplicity, the image signal is normalized to range $[0, 1]$ (where 0 and 1 correspond to black and white, respectively). Thus, the signal represents a perfectly sharp black-to-white edge. Its Fourier transform [10]

$$U(\omega) = \mathcal{F}\{u(x)\} = \pi \delta(\omega) + \frac{1}{j\omega}$$

contains the Dirac function $\delta(\cdot)$, which equals one if its argument equals zero, and equals zero otherwise. The Fourier transform of $u(\cdot)$ has a magnitude $|U(\omega)| \sim 1/\omega$ when ignoring the $\delta(\cdot)$ function. Ignoring the Dirac function is reasonable because $\delta(\cdot)$ represents the frequency coefficient for $\omega = 0$, which is not relevant to the maximum frequency we are looking for.

The spatial signal $g(x)$ is a shifted version of $u(x)$, which corresponds to a multiplication of $U(\omega)$ with the complex exponential $e^{-j\omega x_0}$ [10]. This process does not change the magnitude of the original frequency spectrum of $u(x)$:

![Fig. 2: The input signal $g(x)$ moves right with velocity $v$. Convolution with camera filter $h(x)$ creates the blurred output signal $g'(x)$. Camera filter $h(x)$ actually is a temporal filter, but in this example represented by its spatial equivalent for simplicity. Filter width $d$ equals exposure time $t_{exp}$ multiplied by speed $v$, see Eq. (6).](image-url)
\[ |G(\omega)| = |\mathcal{F}\{g(x)\}| = |\mathcal{F}\{u(x)\}| \cdot |e^{-j\omega x_0}| \\
= |\mathcal{F}\{u(x)\}| = |U(\omega)| \sim \frac{1}{\omega}, \tag{5} \]

so it is still proportional to the reciprocal of \( \omega \). Signal \( g(x) \) moves to the right at velocity \( v \), as indicated in the top left graph in Fig. 2. This process is recorded by a camera with exposure time \( t_{\text{exp}} \). Consequently, during the exposure time, \( g(x) \) moves
\[
d = v \cdot t_{\text{exp}} \tag{6} \]
to the right. The resulting, blurred signal \( g'(x) \) is depicted in the bottom graph of Fig. 2. Note that at each location \( x \), the signal is integrated over time. At locations \( x < x_0 \), input \( g(x) \) will equal to zero at all times during exposure. At locations \( x > x_0 + d \), the input will equal one at all times during exposure. In these locations, \( g(x) = g'(x) \) holds. At \( x = x_0 \), the signal on the (one-dimensional) camera sensor is for all times during exposure equal to zero, except for an infinitesimally small time at the very beginning. Thus, the resulting value in the blurred image in this location equals zero. At \( x = x_0 + 0.1 \cdot d \), the signal is equal to one for ten percent of the exposure time, zero, for the remaining 90 percent. Therefore, the resulting image value equals 0.1. The linear cascade spans the entire exposure time area, yielding the ramp that can be seen in Fig. 2. This linear dependency holds for every camera exposure process of an object moving at constant velocity.

The previous insight allows us to formalize how temporal integration during camera exposure affects the spatial frequency of an input signal \( g(x) \) moving at constant velocity \( v \). To simplify the following proof, we replace temporal integration of \( g(x) \) during exposure time by the convolution \( g(x) * h(x) \). The convolution needs to yield the same \( g'(x) \) as temporal integration. To this end, we design the dimensions of \( h(x) \) relative to movement speed \( v \) and exposure time \( t_{\text{exp}} \) using Eq. (6):
\[
h(x) = \frac{1}{d} (u(x) - u(x - d)) \tag{7} \]
The convolution \( g(x) * h(x) \) will always lead to the same \( g'(x) \) as temporal integration with exposure time \( t_{\text{exp}} \) of a signal \( g(x) \) moving at constant velocity \( v \). Filter \( h(x) \) is shown in the top right graph in Fig. 2. The magnitude of its Fourier transform
\[
|H(\omega)| = |\mathcal{F}\{h(x)\}| = \frac{1}{d} |\mathcal{F}\{u(x)\}| \cdot |1 - e^{-j\omega d}| \tag{8} \]
equals the magnitude of \( U(\omega) \) multiplied with the inverse of \( d \) and the magnitude of \( 1 - e^{-j\omega d} \). Let us derive
\[
c(\omega, d) = |1 - e^{-j\omega d}| \tag{9} \\
= |1 - \cos(\omega d) + j \cdot \sin(\omega d)| \\
= \sqrt{(1 - \cos(\omega d))^2 + (\sin(\omega d))^2} \tag{10} \\
= \sqrt{1 - 2 \cos(\omega d) + (\cos(\omega d))^2 + (\sin(\omega d))^2} \tag{11} \\
= \sqrt{2 - 2 \cos(\omega d)} \in [0, 2]. \tag{12} \]

It is important to notice that \( c(\omega, d) \) is a periodic function in the domain \( c(\omega, d) \in [0, 2] \). \( c(\omega, d) \) will therefore not play a significant role for the following proportionality considerations. Consequently, the magnitude from Eq. (8)
\[
|H(\omega)| = \frac{c(\omega, d)}{d} |U(\omega)| \sim \frac{c(\omega, d)}{\omega d} \tag{14} \]

approximates \( \frac{1}{2} |U(\omega)| \), and, more importantly, is approximately proportional to the reciprocal of the product \( \omega d \) because \( c(\omega, d) \in [0, 2] \). Thus, the filter width \( d \), which is a function of \( t_{\text{exp}} \) and \( v \), see Eq. (6), defines how quickly the frequency coefficients of \( |H(\omega)| \) will converge to zero. Finally, we retrieve the magnitude spectrum of the output signal \( g'(x) \):
\[
|\mathcal{F}\{g'(x)\}| = |\mathcal{F}\{g(x) \ast h(x)\}| \\
= |\mathcal{F}\{g(x)\}| \cdot |\mathcal{F}\{h(x)\}| \\
= |G(\omega)| \cdot |H(\omega)| \\
= |U(\omega)|^2 \frac{c(\omega, d)}{d} \left( \frac{\omega d c(\omega, d)}{\omega^2} \right) \tag{18} \\
= \pi \delta(\omega) + \frac{1}{j\omega} \left( \frac{\sqrt{2 - 2 \cos(\omega d)}}{d} \right) \tag{19} \\
= \frac{1}{\omega^2} \left( \frac{\sqrt{2 - 2 \cos(\omega t_{\text{exp}} v)}}{t_{\text{exp}} v} \right) \forall \omega \in \mathbb{R} \setminus \{0\} \tag{20} \]

The above calculation uses the convolution theorem [10] to transform line (15) to line (16), and Eq. (5) and (14) to convert line (17) to line (18). In line (18), we observe that the magnitude spectrum of output signal \( g'(x) \) decreases faster than the input signal at higher frequencies \( \omega \) because of its approximate proportionality to the reciprocal of \( \omega^2 d \) because of \( \omega^2 t_{\text{exp}} v \), which gives it the blurred appearance. Still, the magnitude spectrum does not generally vanish at an upper limit \( \omega_0 \). This is why result (3) cannot be used directly on real signals and further investigations in the remaining sections of this paper need to be done.

We express the spectrum magnitude of output \( g'(x) \) as a function of exposure time \( t_{\text{exp}} \) and speed \( v \) in Eq. (20) to give a better intuition of how these two variables influence the spatial frequency components. As we know from Eq. (13), the numerator in Eq. (20) can take values in \([0, 2]\). Therefore, the spectrum magnitude of \( g'(x) \) is approximately proportional to, among others, \( 1/t_{\text{exp}} \). Thus, e.g., a longer exposure time \( t_{\text{exp}} \) will lead to faster decay of frequency components of the magnitude spectrum of \( g'(x) \), allowing a smaller critical sampling frequency \( f_c \).

The discussion in the previous paragraphs assumed infinitely high spatial resolution. In real systems, intensity signals are spatially sampled by the pixel matrix in the camera sensor. This leads to a staircase function instead of a ramp function for \( g'(x) \) in Fig. 2. The staircase signal would be composed of unit step functions, yielding a spectrum magnitude that is vanishing slower than in Eq. (18), hence causing a larger critical sampling frequency \( f_c \). However, at normal viewing conditions, state-of-the-art displays have pixel densities of usually 35 to 150 pixels per degree (an overview is provided in Tab. II). This is close to and often greater than the angular...
resolves the human eye, approximately 0.02 degrees with optimal eyesight [11], corresponding to 50 pixels per degree. Hence, the spatial discretization effect is negligible.

Our derivations in this section serve three purposes: first, they show that there is no upper bound on the spatial spectrum of an output image created with a conventional camera. Second, they introduce the reader to how the camera exposure process filters an input scene containing motion. Third, they provide insight into the measurement of motion blur. This section describes the presented sequences, their creation and presentation, as well as the experimental procedure. Results from these experiments are going to verify through experiments in Section III.

C. Contributions

As argued in Section I-B, hard cut-off thresholds usually do not exist in real signals, making additional tests on top of Watson’s work in [9] necessary. We proceed in Section II with the description of our experiment including presented sequences, their creation and presentation, as well as the experimental procedure. Results from these experiments are presented in Section III, and discussed in Section IV. We have found that both, the speed of a moving object and the exposure time of the camera recording the object, have a statistically significant influence on the critical sampling frequency $f_c$. We create a model to determine $f_c$ for a given object speed and camera exposure time and propose a formula to compute the object speed from an encoded video sequence.

A. Sequences and Sequence Parameters

As it is common practice in psychometrics, we present an object moving in one dimension to the users [4], [8], [9]. Our sequences are shown on a special computer monitor, which is described in more detail in Section II-C. The shown sequences, for an example see Fig. 3, contain a vertical white bar that spans the entire display height, and one fifth of the display width. The white bar moves from left to right over black background at constant speed. This setting gives us maximum contrast and clearly perceivable intensity edges. When general video content is presented to users, lower contrasts might be present, which would require lower frame rates. Therefore, our results will serve as an upper bound for the critical sampling frequency $f_c$.

The bar starts at the left side of the monitor, and when it moves out of the display on the right side, it moves into the display again on the left side. We compute the moving bar with a simulation rate of 20 kHz. This allows us to simulate the relevant exposure times for the camera and the resulting motion blur. The simulation of exposure time is described in Section II-B. The blurring caused by exposure time can be seen at the edges of the white bar in Fig. 3.

For each set of sequences, we set a constant speed $v$ for the bar, and a constant exposure time $t_{exp}$ for the camera. Each set contains sequences with frame rates $f_{seq}$ ranging from 10 Hz to 90 Hz in steps of 2 Hz. For referencing, we label the sets in Tab. I. In this table, the speed and exposure time increase from top to bottom: first, the bar is starting at the left side of the monitor, second after moving half way to the right, and third when the bar is crossing the right display border to reappear at the left border. In this figure, each image is shown at 1/3 vertical height.

Fig. 3: Sequence C (see Tab. I) at three time instances during playback (from top to bottom): first, the bar is starting at the left side of the monitor, second after moving half way to the right, and third when the bar is crossing the right display border to reappear at the left border. In this figure, each image is shown at 1/3 vertical height.
be 48.3 deg/s (i.e., sequence sets A, B and C). We did not choose 72.5 deg/s as at such a high speed, the bar is passing the display width so quickly that it is extremely exhausting for test subjects to register the presence or absence of jitter.

We chose the parameter limits based on the following considerations: in conventional modern video applications, videos are recorded at frame (image) rates between 30 Hz and 60 Hz. This yields sampling periods between 33.3 ms and 16.7 ms, which serve as an upper bound for the exposure time because exposure of one image in a video may not take more time than the corresponding sampling period. We are in addition particularly interested in short exposure times of high frame rate videos with around 200 Hz of frame rate, as used in low delay video communication such as [2]. Therefore, we set the exposure range to [5.0, 25.0] ms to cover current and future applications.

The upper speed limit is decided by what participants were able to judge with reasonable effort. It is observed that higher speeds are difficult to follow given that the display covers only 46.4 degrees of the subject’s field of view (FoV). The lower speed bound is dictated by what participants were able to perceive with reasonable effort. In this case, test subjects will directly perceive temporal or spatial sampling frequencies are lower than human perceptual thresholds, test subjects will directly perceive temporal or spatial sampling.

### B. Sequence Creation

A Python script creates the sequences using OpenCV [12], encodes the resulting video in h.264 [13] and writes it to a file for later presentation to the users. The script simulates the movement of the vertical bar at speed \( v \) in discrete-time steps at \( f_{\text{sim}} = 20 \text{kHz} \). This means that the script creates an image showing the bar at its current position 20,000 times per second in simulation time, and then applies the camera exposure time filter to create the output video sequence with a given frame rate, object speed, and exposure time. The simulation frequency was chosen considering that it provides a sufficiently small simulation complexity (all sequence sets could be created in approximately three days) and imperceptibly small time discretization effects. At \( f_{\text{sim}} = 20 \text{kHz} \), the temporal offset of capturing an event at an exact simulation time is at most 25 µs, which proved to be small enough for not being perceived by the test subjects. At lower simulation frequencies, time discretization effects become visible in some sequences.

The discretization effect can influence the resulting sequence in two perceivable ways: first, if temporal or spatial sampling frequencies are lower than human perceptual thresholds, test subjects will directly perceive temporal or spatial sampling. Second, imagine a bar that moves to the right 1.01 pixels per simulation step. After spatial discretization, its position will be incremented by one pixel, except for every 100th step, at which the position is incremented by two pixels. In this case, test subjects will be exposed to flicker with a frequency which is 100 times smaller than the simulation frequency. The analogous insight holds for the discretization of time.

In the temporal domain, this discretization effect is attenuated below the human perceptual threshold by choosing a large enough simulation frequency of 20 kHz. However, in spatial domain, we are limited by the resolution of our display panel. This proved to be insufficient, which is the reason why we added linear interpolation at the edges of the bar: imagine that the left edge of the bar should be drawn at pixel position 2.4 (between pixels 2 and 3). Without interpolation, pixel 1 would be black (value 0), and pixels 2 and 3 would be white (value 255). With linear interpolation, pixels 1 and 3 retain their value, but the value of pixel 2 equals to \( (1 - 0.4) \cdot 255 = 153 \), where the term \((1 - 0.4)\) generalizes to “One minus distance to the closest integer pixel position”. The use of linear interpolation reduced spatial discretization below human perception thresholds. In general, the degree of interpolation of pixel values has to equal the degree of the equation for object location with respect to time. A linear acceleration for example would require quadratic interpolation. Other movements could be interpolated using approximation through splines. As in our experiment, the white bar is moving at constant velocity (linear motion through space), linear interpolation is sufficient.

Finally, the script needs to simulate motion blur caused by the exposure time of the camera and the movement of the bar. This is done in a rather simple way: the script always keeps the bar images from the past \( k = f_{\text{sim}} \cdot t_{\text{exp}} \) simulation steps. In order to create an output image of the camera, the script...

---

**TABLE 1:** Labels for the used sequence sets with various speeds \( v \) [degree/second] and exposure times \( t_{\text{exp}} \) [milliseconds]. Two authors of this paper performed psychophysical tests on all sets, the remaining subjects only on sequence sets with bold letters and gray background.

<table>
<thead>
<tr>
<th>( v ) [deg/s]</th>
<th>5 ms</th>
<th>10 ms</th>
<th>15 ms</th>
<th>20 ms</th>
<th>25 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.5</td>
<td>I</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>48.3</td>
<td>A</td>
<td>O</td>
<td>B</td>
<td>P</td>
<td>C</td>
</tr>
<tr>
<td>36.3</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
</tr>
<tr>
<td>24.2</td>
<td>D</td>
<td>V</td>
<td>E</td>
<td>W</td>
<td>F</td>
</tr>
<tr>
<td>18.1</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>12.1</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
</tr>
<tr>
<td>6.0</td>
<td>G</td>
<td>h</td>
<td>H</td>
<td>i</td>
<td>I</td>
</tr>
</tbody>
</table>
computes the mean image of the past $k$ images, which equals convolving the input image with the camera filter $h(x)$, see Fig. 2.

C. Sequence Playback

This section discusses the sequence player, the display, and the physical experimental setup. The player should be able to play the high-resolution sequences from the Python script (Section II-B) with precise frame timing. No available video player could meet this requirement, so we created our own player. It uses libav\(^2\) for decoding, and SDL\(^3\) for displaying. The decoder can decode any frame from any sequence within 1.5 ms, which is much smaller than the minimum frame period (i.e., 11.1 ms) we have in our sequences.

In the following, we assume that we want to display a video at a frame rate $f_s$ with frame period $t_s = 1/f_s$. For frame timing, we used the `wait_until()` function of the chrono library. Just before showing a frame/image on the screen, we record the current time in time stamp $t_0 = t$. Next, we update the screen content and perform all computationally intensive and time varying tasks such as file reading, decoding, color space conversion and copying to the GPU. Then we wait until $t_0 + t_s$, set the new time stamp to the current time $t_0 = t$ and update the screen content with the newly decoded image. This process is repeated until there is no frame left in the video file.

Because of processing overhead from setting the time stamp, the actual frame rate of the player was, depending on the frame rate, up to 1 Hz lower than the target frame rate $f_s$. This has been fixed by reducing $t_0$ (increasing the frame rate) in the player according to a regression based on previous playback timing data. With that, the average playback frame rate did not deviate more than 0.05 Hz from the target frame rate. In addition, we evaluated the playback frame periods, and found them to closely match $t_s$, with a maximum absolute deviation of 30 $\mu$s per frame period.

We used an Acer XB270H as display. It has a 27-inch TN-panel, with a resolution of 1920 horizontal times 1080 vertical pixels. It supports Nvidia G-Sync, which refreshes the image whenever a new frame is available in the frame buffer (i.e., in contrast to conventional monitors) up to a maximum refresh rate of 144 Hz. With our sequences, we do not exceed 144 Hz, so this display allows us to show videos with the exact frame rate defined by our video player. Conventional monitors have fixed refresh rates, leading to delayed presentation of some video frames.

We evaluated both the player and the display time precision by encoding a video sequence that showed black and white frames interchangeably. While playing back this sequence, we put a photoresistor on the display surface and sampled its resistance with $8$ kHz to observe the brightness change of the monitor with a high time resolution. This experiment proved that the setup is working as expected, frame period variations were smaller than $1/8$ kHz $= 125$ $\mu$s.

The width of the display panel is $w = 60$ cm, test subjects are seated with an eye-to-panel distance of approximately $d = 70$ cm in front of the monitor. Therefore, the display spans 46.4 degrees of the horizontal field of view of the test subjects. Lighting in the room is dominated by artificial light to eliminate the influence of daylight or weather conditions.

D. Participants

In addition to the two authors, seven male and one female subject were recruited to participate in the experiment. The age of all participants at experiment time ranged from 23 to 30 years, with a mean of 27.6 years. All participants have either normal or corrected-to-normal eyesight.

Fig. 4: Example of an adaptive interleaved staircase test with two sequence sets. In each round, set I (black line, absolute frame rate threshold $f_{c,1}$) or set g (orange line, absolute frame rate threshold $f_{c,g}$) is chosen randomly. Within each set, the answer to a sequence (x: could perceive stutter, o: could not perceive stutter) increases (x) or decreases (o) frame rate $f_{seq}$ of the next sequence from that set by the current step size. Step size is 10 Hz in the beginning, and reduced to 2 Hz after four reversals in that set.

E. Experimental Procedure

In preliminary experiments, we used the method of constant stimuli [14] to retrieve a prior estimate of the absolute frame rate threshold, equal to the critical sampling frequency $f_c$ from Section I-A, at which humans can distinguish time-discrete presentation of a video sequence from the continuous presentation of the same scene 50 $\%$ of the time. In the method of constant stimuli, test subjects are randomly presented a stimulus (video sequence) from a range of stimulus parameter values, in our case from a frame rate range. Stimuli from the range have to be sampled with equal spacing, and each stimulus must have equal probability for presentation to avoid errors of habituation and expectation. Hence, this method frequently presents stimuli which are rather distant from threshold of habituation and expectation. Hence, this method frequently presents stimuli which are rather distant from threshold $f_c$ and accordingly easy to classify (redundant).

To avoid this issue, we used a modified variant of the simple up-down or staircase method [15], [16]. In the up-down method, the next sequence is chosen based on the answer for the previous sequence: if jerkiness was perceived (answer x in Fig. 4), the frame rate of the next sequence will be increased by for example 2 Hz, if no jerkiness was perceived (answer o in Fig. 4), frame rate will be lowered by the same step size of

\(^2\)https://www.libav.org/, last visited 03.09.2018

\(^3\)https://www.libsdl.org/, last visited 03.09.2018
2 Hz, see the black line at test round number 13 and greater in Fig. 4. This procedure ensures that frame rates close to threshold $f_c$ will be frequently sampled, avoiding redundant stimuli. However, test participants will be able to predict frame rate, which leads to errors of expectation.

This is why we used the interleaved staircase method [16]. Here the test subject is presented, e.g., sequence sets I and g (see Tab. I for parameters, and Fig. 4 for an example). For each sequence set individually, the simple staircase method is used. In addition, the sequence sets are randomly interleaved, meaning that the next sequence to be presented is randomly chosen from the sequence sets I and g. After choosing the set of the next sequence, the sequence frame rate $f_{\text{seq}}$ is determined according to the simple staircase based on the previous answer for the chosen set. This strongly impedes predicting frame rate, in particular if the number of sets is large enough and if the set parameters speed and exposure are sufficiently similar. In our experiments, we used at least three sequence sets for interleaving, making it impossible for test subjects to predict frame rates.

Further potential in increasing the efficiency of our experiment lies in its beginning: we start at a low [16] frame rate $f_{\text{seq}} = 20$ Hz, which all test subjects classified as jerky for all sequence sets. The following considerations apply to one set: during the test, participants first have to converge to the absolute frame rate threshold $f_c$ before being able to give the most relevant answers.

To speed up convergence, a step size larger than, e.g., 2 Hz could be used, which would decrease the precision of the result. Therefore, we use an adaptive step size in two experiment phases [17], [18], as shown in Fig. 4: in the first phase, we prefer to quickly converge to $f_c$, and thus choose a large step size of 10 Hz. This phase terminates when the test subject’s answers have caused four reversals. A reversal takes place when the chosen frame rates $f_{\text{seq}}$ pass through an extremum, meaning that frame rate was first decreased and then increased, or vice versa. After four reversals, the second phase, in which we use a smaller step size of 2 Hz, starts. The experiment finishes when twelve reversals have taken place during the second phase in each of the sequence sets.

We gave an oral introduction to all test subjects. In addition, they received an introductory sheet, which described the experiment, the user interface and encouraged behavior for result consistency, such as keeping a constant distance to the monitor (not leaning towards or away from it) and taking breaks if needed. The sheet states that “sequences are randomly chosen without any correlation”, concealing the underlying adaptive interleaved staircase method. Distinguishing monocular from binocular vision is not necessary for motion perception [3], so we did not take any precautions in this respect.

The 35 sequence sets of the two author subjects from Tab. I were divided into eight interleaved experiment blocks with four sequence sets each, and one experiment block with three sequence sets. The nine sequences of the remaining eight test subjects, highlighted in Tab. I, were divided into one experiment block with the four sequence sets A to D, and one experiment block with the five sequence sets E to I.

III. RESULTS

The participants’ answers were used to compute an absolute frame rate thresholds (critical sampling frequencies) $f_c$. Corner points of the tetragons that constitute the bent planes correspond to the tested sequence sets and their absolute frame rate threshold $f_c$. The horizontal layout of the sample points in the velocity/exposure plane is based on Tab. I. Box plots corresponding to Fig. 5b are depicted in Fig. 6.

A. Average Behavior of the Author Test Subjects

First, we conducted the experiment on all 35 sequence sets with the two author test subjects. The resulting $f_c$ for all sets are depicted in Fig. 5a. In the horizontal layout of the graph in Fig. 5a, the array of sequence sets from Tab. I can be recognized.

We do not give a more precise plot or more values of $f_c$ because this experiment is motivational and its exact values do not generalize, since it was conducted on only
two users. Still, the experiment shows that test sequence set
sub-sampling is feasible. Revisiting the four hypotheses from
Section I-B, we observe the following: first, we can see that
higher speed \( v \) increases the absolute frame rate threshold \( f_c \)
(hypothesis 1). Compare, for example, \( f_{c,D} = 59.83 \text{ Hz} \), which
shows the moving bar at an angular speed of \( 72.5 \text{ deg/s} \) and \( f_{c,G} = 40.92 \text{ Hz} \), which has an angle speed of \( 6.0 \text{ deg/s} \) (see
Tab. I). Hence, from the two opposing effects of speed \( v \) on \( f_c \)
presented in Section I-B, the effect changing the steepness
of the spatiotemporal spectrum (Fig. 1 and Eq. (3)) has more
influence than the filter width \( d \) (Eq. (6) and Eq. (18)).

The second hypothesis, stating that higher exposure will
decrease \( f_c \), holds only for large enough speeds. There is a
6 Hz difference between \( f_{c,3} \) and \( f_{c,N} \), but only a 0.84 Hz
difference between \( f_{c,G} \) and \( f_{c,1} \). The decrease in influence is
understandable because the width of the camera filter \( h(x) \)
is equal to the product of speed and exposure time. Also, this
insight confirms the third hypothesis, stating that at reduced
speed the influence of exposure time will diminish.

Finally, the difference between \( f_{c,3} \) and \( f_{c,G} \) is larger than the
difference between \( f_{c,N} \) and \( f_{c,1} \), confirming the fourth
hypothesis: longer exposure time (causing increased motion
blur) decreases the influence of speed.

All previous insights also hold for all test subjects, as we
show in Section III-B.

B. Average Behavior of All Test Subjects

The graph in Fig. 5b illustrates the results for the average
absolute frame rate threshold \( f_c \). It can be seen that the
conclusions from Section III-A generalize to all test subjects.
The tendencies of frame rate thresholds of individual participants
match the tendencies in Fig. 5b well; corresponding box plots
are given in Fig. 6. We also see a grouping effect: sequence
set D was presented in an interleaved experiment together with
sets A, B, and C. This results in a comparably high \( f_{c,D} = 51.68 \text{ Hz} \), in contrast to \( f_{c,E} = 46.58 \text{ Hz} \) from sequence
set E, which was interleaved with sets F to I.

Section IV-A gives a statistical analysis of the results.

C. Test Length and Difficulty

The ten test subjects spent overall 6 hours and 40 minutes
performing the experiments for sets A to I, giving 2734
answers. Experiment time per participant was between 25 and
61 minutes, with an average of 44 minutes. For each sequence,
participants spent on average 8.8 seconds from starting to
watch the sequence until submitting an answer (fastest particip-
\( t \) on average 5.8 seconds, slowest participant: on average
15.6 seconds).

Users took on average two seconds more per sequence on
the first experiment block with sets A to D than on the second
block with sequence sets E to I. This is understandable since
they started with the first block and had to get used to the
experiment. Otherwise, we found no statistically significant
difference in the average time users spent on classifying one
set, or the average number of answers given for a set. Given
no statistically significant difference, we can say that for
none of the chosen sequence set parameters, test subjects had
difficulties to complete the experiment.

D. Directional Independence

Existing literature does not observe a dependency between
the direction of motion and the corresponding absolute frame
rate threshold \( f_c \). Cox and Derrington [19] presented eight
motion directions to participants and found only minor and
inconsistent differences for the temporal frequency threshold
for a motion signal to be perceivable in contrast to no motion.
The authors in [7] use a rotating disk for presenting motion in
their experiments and do not report any direction influence.

To confirm these results, we rotated our screen 90 degrees,
such that the bar was moving from top to bottom, and
performed the tests from Section II-E on one participant.
As expected, the differences to the thresholds \( f_c \) from the
horizontal experiment were small and inconsistent. Thus,
motion direction does not change the absolute frame rate
perception threshold \( f_c \). We use this result for simplifying the
computation of required video frame rates in Section V-B.

IV. DISCUSSION

In this paper, a constant sampling frequency (CSF) system
denotes a system which samples signals with a constant period
of time between two successive samples. Therefore, sampling
with constant frequency is, in this manuscript, also temporally
uniform sampling. In contrast to that, adaptive sampling fre-
quency (ASF) systems do not have a fixed sampling period.
The period between two successive samples is adapted to the
sampled data.

To make notation more consistent, we consider sampling
rates or frequencies for cameras as well as for displays. The
notation is clear for cameras, which sample the incoming light.
On the other hand, the data update process in displays is
typically called image refresh, and thus refresh rate or refresh
frequency describe the display update process. Actually, in
display refresh, the display samples the contents of the frame
buffer of the graphics processing unit that the display is
attached to. This is why it is correct to call the display refresh
process a sampling process and accordingly call the refresh
rate of a display a sampling frequency.

In summary, we refer in this paper to CSF and ASF cameras
as well as CSF and ASF displays.

A. Statistical Significance of Tendencies

We analyze the statistical significance of the results from
Section III. For a better understanding, box plots of the
absolute frame rate thresholds \( f_c \) are depicted in Fig. 6. In
Fig. 5b we see that increasing speed seems to increase the
absolute frame rate threshold \( f_c \) for each of the exposure
times 5 ms, 15 ms, and 25 ms. Applying Fisher’s one-way
analysis of variance (ANOVA) test [20] to the results from
sets A, D, and G (\( t_{\text{exp}} = 5 \text{ ms} \)) reveals significant differences
\( F(2, 27) = 37.24, p < 0.001 \) of the thresholds. We find
similar results \( F(2, 27) = 32.83, p < 0.001 \) for sets B, E,
\( t_{\text{exp}} = 15 \text{ ms} \) and for sets C, F and I (\( t_{\text{exp}} = 25 \text{ ms} \)), in which the ANOVA values \( F(2, 27) = 26.65, p < 0.001 \)
are once more statistically significant. The post-hoc Tukey
test further revealed that the mean threshold of one set is
statistically significantly different \((p < 0.01)\) from the other two set means for each exposure time.

As can be seen in Fig. 6, the influence of exposure time on threshold \(f_c\) is smaller than the effect of speed. Observing the box plots from sets A, B, and C, we also see that the variance of thresholds between sequence sets is small compared to the variance of thresholds within sets. Nevertheless, for the majority of test subjects, we saw a strong and consistent correlation between the individual test subject thresholds and exposure time \(t_{\text{exp}}\). Therefore, the subjective differences between test subjects would conceal the existing dependency of threshold \(f_c\) on exposure time \(t_{\text{exp}}\) in a classical ANOVA test. This is why we apply a repeated measures ANOVA (RANOVA) test \([20]\) to check statistical significance of the effect of exposure time \(t_{\text{exp}}\) on \(f_c\). The RANOVA test computes the statistical significance of difference among data sets while ignoring variations between test subjects.

Applying RANOVA to sets A, B, and C yields significant differences \((F(2,18) = 29.69, p < 0.001)\) for thresholds \(f_c\) for varying exposure times at speed \(v = 48 \text{ deg/s}\). In addition, the post-hoc Tukey test reports significant pairwise differences for all set mean thresholds. The difference is significant at the 0.01 level in set pair (A,B) and (A,C) and significant at the 0.05 level in set pair (B,C). For sets D, E, and F, RANOVA \((F(2,18) = 17.05, p < 0.01)\) again yields significant differences, but the post-hoc Tukey test finds significant difference only on the pairs (D,E) and (D,F) at the 0.01 level. This result has to be read with caution, since the threshold result \(f_{c,D}\) is higher than what we expected because of the grouping effect discussed in Section III-B. Finally, for \(v = 6 \text{ deg/s}\) on sets G, H, and I, RANOVA \((F(2,18) = 3.19, p = 0.065)\) reveals that statistical significance at the 0.05 level is just missed. To back up this result, the post-hoc Tukey test finds no statistically significant differences in all three set pairs.

In summary, the tests confirm hypothesis 3: exposure time significantly affects \(f_c\) at high speeds \(v\), while the influence vanishes at lower speeds.

### B. Conclusions on the Absolute Frame Rate Threshold

In the previous sections, we found that both speed and exposure time have a significant influence on the absolute frame rate threshold, rendering Fig. 5b a valid model for predicting \(f_c\). To be able to apply these insights to general scenarios, we fit a linear function to the data from Fig. 5b:

\[
f_c(t_{\text{exp}}, v) = 40.44 - 153.2 \cdot t_{\text{exp}} + 0.3230 \cdot v,
\]

where \(t_{\text{exp}}\) is given in milliseconds and \(v\) in degree per second. We chose a linear model as with higher degrees for the polynomial we can achieve a closer fit of the model to the values present in Fig. 5b, but the higher degree polynomial does not extrapolate well to values of \(v\) and \(t_{\text{exp}}\) outside the region investigated in our experiment (overfitting). The linear model is based on the mean \(f_c\), and hence critical sampling frequencies computed by it will make temporal sampling imperceptible to 50% of users. To satisfy a greater fraction of users, we can add a frequency offset \(f_o\) to Eq. (21), as done in Algorithm 1.

### C. Mapping the Absolute Frame Rate Threshold to Constant Sampling Frequency Devices

Nowadays, in the vast majority of used displays, the constant sampling frequency (CSF) is not synchronized to the image source, in contrast to the specific display used in our experiment from Section II. In addition, practically all cameras in use (except dynamic vision sensors \([21]\)) have constant sampling frequencies. Hence, we show how to map the frame rate threshold \(f_c \in \mathbb{R}^+\) to its quantized version \(\hat{f}_c\), dictated by the sampling frequencies of the involved CSF devices. The effective sampling frequency \(f_s\) of images visible at the display is equal to the minimum of the involved sampling frequencies. Assume for example a 50 Hz camera and a 60 Hz display, or the other way around: the effective displaying rate will in both cases be 50 Hz, as the device with higher sampling frequency will sample at least once during the sampling period of the lower sampling frequency device. The example shows that it does not matter whether the camera or the display is sampling at a lower frequency. For the following thought experiment, we assume the display to have the lower, constant sampling frequency.

A constant sampling frequency display can show videos with the original sampling rate \(f_s\) and integer fractions \((f_s/2, f_s/3 \text{ etc.})\) of \(f_s\) by skipping one or more new images. Fig. 7 shows the possible reduced display sampling frequencies for \(f_s \in [30,350] \text{ Hz}\). As target displaying frequencies for this thought experiment, we take the range of \(f_c \in [40,60] \text{ Hz}\) from Fig. 5 plus/minus a plausible safety margin of 20 Hz. Therefore, we target displaying frequencies in the range from 20 Hz to 80 Hz. We are in this example not interested in

---

**Fig. 6:** Box plots of all participant thresholds for each of the nine sequence sets A to I. The boxes contain 50% of the data, the horizontal lines in them represent the median. Whiskers are at most 1.5 times the box height (interquartile range) and end at the lowest or highest found threshold value. Pluses denote outliers.
displaying higher frequencies than 80 Hz, as users will probably not perceive the increase in frame rate over 80 Hz, and we are also not interested displaying a video below 20 Hz, as humans will certainly perceive temporal sampling at that frequency. It can be seen in Fig. 7 that for low $f_s = 50$ Hz we can only choose between two target frequencies (50 Hz and 25 Hz) in the relevant range between 20 Hz and 80 Hz, while for increased sampling frequencies there are more displaying frame rates available (e.g. for $f_s = 80$ Hz, these are 80 Hz, 40 Hz, 26.7 Hz, and 20 Hz). In Section IV-D, we analyze how this frame rate quantization deteriorates the frame rate reduction that can be achieved by using the adaptive sampling rate proposed in this paper.

Since the goal is that the user is unable to perceive the sampling process, we want to display the video with the absolute frame rate threshold $f_c$ or a higher frequency. As an example, let us assume that we have a display with 120 Hz, and the video content characteristics demand that it is shown at $f_c = 50$ Hz. With the given display, we can display 120 Hz, 60 Hz, 40 Hz and so on. Thus, we choose to display it with the frame rate just above $f_c$, which is in this case $f_c = 60$ Hz. The analog thought experiment can be done for a CSF camera. We have implemented this temporal quantization in Section V-C. Fig. 9 shows the trace of the desired $f_c$ for a video sequence and the actually displayed quantized frame rate $\hat{f}_c$.

**D. Reduction of the Number of Frames in Constant Sampling Frequency Devices**

As can be seen in Fig. 9, CSF systems require the video sequence to be displayed at a higher frequency $f_c$ than requested by Eq. (21). In this section, we analyze the number and ratio of additional frames needed to show the video when a constant sampling rate device is involved.

**V. APPLICATIONS IN VIDEO COMPRESSION**

The insights from Section IV can be applied to video compression. To do so, we need to know pixel densities of various display classes, so an overview is given in this section. Subsequently, we present the procedure of extracting the adaptive playback frame rate from a coded video sequence. Finally, we provide experimental results of the proposed technique.

---

Fig. 7: Possible display sampling frequencies $f_{s,\text{trace}}$ for given original sampling frequency $f_s$. Available display sampling frequencies are highlighted for $f_s = 50$ Hz and for $f_s = 80$ Hz. For clarity, we show the legend only for the first seven integer fractions of each $f_s$. Greater fractions are depicted by gray lines.

Fig. 8: Ratio of required frames $n_c$ for playback on CSF displays compared to the number of frames required for playback on ASF displays $n_a$. Note that for $f_s < \bar{f}_c = 15$, users will be able to perceive stutter because in this model, the maximum occurring $f_c$ is in that case greater than $f_s$.
A. Display Classes

Various display classes have different pixel densities $\rho$. In our overview, we only cover horizontal size and angle, since the vertical size as well as angle directly follow for a given aspect ratio. Hence, knowing the size in one dimension allows us to map motion given as pixel offset to motion in centimeters per second on the screen. Typical display dimensions, distance of the user to the display, user’s FoV covered by the display, the number of pixels in a display row, and the pixel density $\rho$ from user perspective are given in Tab. II. In this table, we see that even within one display class pixel density can vary widely. The values in the last column of Tab. II can be used in Eq. (23) to map the on-screen speed to an angular velocity $v$ from a spectator’s perspective.

B. Extracting $f_c$ from Coded Video Sequences

From motion vectors of a coded video sequence (in our example the h.264 [13] video codec), we can obtain the absolute frame rate threshold $f_c$ for each video frame. The procedure is summarized in Algorithm 1.

\begin{algorithm}
\caption{Frame rate determination for frame $i$}
\begin{algorithmic}
1: Extract motion vectors $m_{v,k,l,i}$
2: Scale motion vectors based on distance to reference frame
3: Apply 3D median filter (22) to the motion vectors
4: Compute maximum motion vector magnitude $m_i$
5: Calculate speed $v_i$ using Eq. (23)
6: Compute frame rate threshold $f_{c,i}$ using Eq. (21)
7: Add desired frequency offset $f_o$ to $f_{c,i}$
8: Quantize $f_{c,i}$ to an integer fraction of $f_s$
\end{algorithmic}
\end{algorithm}

The h.264 codec identifies a matching block in the reference frame for each block in the current frame. The matching block is later used for prediction. Ideally, the matching block corresponds to the same image content, which has been offset as a result of motion apparent in the video. This is why the vectors indicating the spatial relation between the two blocks are called motion vectors. Occasionally, block similarity is not caused by motion, but just by similarity of image content in different locations. In that case, motion vectors do not represent the true motion. Hence, we need to filter the set of motion vectors.

Given the motion vectors $m_{v,k,l,i}$ at block locations $k,l$ (vertical, horizontal) of frame $i$, we first scale the motion vectors according to the temporal distance to their reference frame. If the reference frame is temporally adjacent, no scaling is done, if the reference is two frames away, the motion vector is divided by two, and so on. Afterwards, we apply the three-dimensional, spatiotemporal median filter (22) to suppress motion vectors that do not represent an actual motion. The filter

\begin{equation}
m_{v,k,l,i} = \text{median}_{m,n,c=\{-2,2\}, j=\{-6,0\}} (m_{v,k+m,l+n,j})
\end{equation}

takes the current motion vector and its spatially as well as temporally neighboring samples as in Eq. (22) as input. These filter dimensions have yielded good results in empirical tests.

The filter only uses past and current motion vector samples to ensure causality.

The 3D median filter is applied to each spatial dimension of the motion vector separately. In the next step, we compute the motion vector magnitude. We are not interested in the direction, as discussed in Section III-D. Finally, for each frame $i$, we compute the maximum motion vector magnitude $m_i$ from all motion vector magnitudes in the frame. This is done for the reason that the maximum motion apparent in the video is going to define the absolute frequency threshold for jitter perceptibility for the person viewing the video. Next, we compute the angular speed $v$ from the user’s perspective.

The maximum motion vector magnitude $m_i$ within a frame divided by the temporal sampling frequency $f_s$ of the video gives us the maximum occurring speed on the display. Together with display pixel density $\rho$ (Tab. II) we can map speed $m_i$ relative to the display panel to an angular speed $v_i$ from the user’s perspective:

\begin{equation}
v_i(m_i, \rho, f_s) = \frac{m_i \cdot f_s}{\rho} \left[\text{deg/s}\right]
\end{equation}

The resulting speed from Eq. (23) can be inserted into Eq. (21) to compute the current critical sampling frequency $f_{c,i}$ for video frame $i$. In the final steps, the frequency offset $f_o$ is added to $f_{c,i}$, and the result is quantized to an integer fraction of $f_s$, as described in Section IV-C.

We have verified the correctness of the algorithm by watching the video and extracting the maximum object offset for a number of frames per hand. Comparing the manually extracted results to the values obtained with the algorithm presented in this section showed that steps 1-4 of Algorithm 1 are working precisely.

An alternative to using the results from video encoder motion analysis is applying a real-time optical flow algorithm [23], [24] to extract motion vectors for each frame. This might be necessary if e.g. video coding parameters restrict the motion vector length.

C. Experimental Results

We cannot apply the proposed methods to widely used test sequences such as the “Foreman” sequence because most of them have been recorded at 30 Hz, which is below $f_c$ for almost all content. Consequently, we created our own sequences: At an exposure time of 2.5 ms, we recorded videos at $f_s = 339$ Hz and a resolution of $960 \times 270$ pixels using a XIMEA MQ022CG-CM camera. The raw image sequences were encoded using the x264 encoder [25], an implementation of the h.264 coding standard [13]. To ensure applicability in low delay video communication scenarios [2], we used the tunings zerolatency and fastdecode as well as preset ultrafast in the encoder.

We recorded sequences in which single or multiple objects are moving in front of a static background and sequences in which the camera is panning at varying speed and direction over three dimensional scenes, yielding various on-screen speeds. In Fig. 9, we can see the resulting $f_c$ (orange) for a part of one sequence. Despite using the ASF display from Section
Section IV-B, we noted that different users require different users view the sequences played back at another available frequency quantization level.

\[ f_{\text{max}} = \frac{339}{3} = 113 \text{Hz} \]

\[ f_{\text{max}} = \frac{339}{4} = 84.75 \text{Hz} \]

\[ f_{\text{max}} = \frac{339}{5} = 67.8 \text{Hz} \]

We verified the model in Eq. (21) and Algorithm 1 by letting us notice any temporal sampling.

The bit rate of a video played back with ASF can be compared to the required bit rate of a video played back with CSF. The constant sampling and display frequency \( f_s \) has to correspond to the highest frequency \( f_{\text{max}} \) in the ASF sequence for humans never to be able to perceive temporal sampling. In the following, we investigate three video sequences recorded with parameters presented in the beginning of this section:

- **MessyRoom**: The camera pans over an untidy room with many details. Camera motion is random, and the camera almost never rests.
- **Whiteboard**: The camera pans over a whiteboard with written text on it, and towards the end of the video, an arm is moving through the camera’s field of view.
- **Catwalk**: The camera is in a fixed position, and a person walks multiple times through the camera’s field of view. The person swings his arms actively, so various objects are moving at different speeds and directions at once.

Tab. III shows the frequency and average bit rate results for these sequences. All videos are encoded at the same default image quality settings to achieve the same (except negligible numerical variations) picture quality for all encoded videos. We investigate ASF Algorithm 1 for frequency offsets \( f_o = 0 \text{Hz} \) and \( f_o = 20 \text{Hz} \). In the corresponding columns, the entries show the average video bit rate \( r \) in megabits per second (Mbps), mean playback frequency \( f_{\text{c}} \), and maximum playback frequency \( f_{\text{max}} \).

For CSF playback, Tab. III shows the average bit rates \( r \) sampling frequencies \( f_s \) corresponding to the maximum playback frequencies in the previous columns. We compare the maximum frequency because in both the ASF and the CSF cases, humans may not be able to perceive temporal sampling. Hence, in the CSF case, the sequence has to be played back at \( f_s = f_{\text{max}} \). This is why we present CSF bit rates only at the frequencies occurring in the ASF cases.

Finally, the last two columns summarize the bit rate savings

\[ \Delta r = \frac{r_{\text{CSF}} - r_{\text{ASF}}}{r_{\text{CSF}}} \] (24)

of ASF over CSF video processing.

We can see that the reduction of average bit rate ranges from 15.4% to 39.1%. The exact amount of rate savings depends mainly on the variance of motion speed \( v \). If variance of speed \( v \) is large, we can expect a greater reduction: ASF would exhibit the greatest bit rate savings if there is high speed apparent for a short period of time (leading to a high \( f_{\text{max}} \).

<table>
<thead>
<tr>
<th>Class</th>
<th>Width [m]</th>
<th>User Dist. [m]</th>
<th>Covered FoV [deg]</th>
<th>Hor. Pix.</th>
<th>Pix. Density ( \rho ) [1/deg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV set</td>
<td>0.8 - 1.3</td>
<td>2 - 5</td>
<td>9 - 36</td>
<td>1280 - 4096</td>
<td>35 - 455</td>
</tr>
<tr>
<td>Smartphone</td>
<td>0.06</td>
<td>0.3 [22]</td>
<td>11</td>
<td>720 - 1440</td>
<td>65 - 130</td>
</tr>
<tr>
<td>Laptop</td>
<td>0.24 - 0.38</td>
<td>0.5</td>
<td>27 - 42</td>
<td>1440 - 4096</td>
<td>71 - 151</td>
</tr>
<tr>
<td>Computer</td>
<td>0.44 - 0.66</td>
<td>0.6 - 0.8</td>
<td>31 - 58</td>
<td>1440 - 4096</td>
<td>25 - 132</td>
</tr>
<tr>
<td>Head-Mounted Display</td>
<td>0.09</td>
<td>0.07</td>
<td>110</td>
<td>1080 - 1600</td>
<td>10 - 15</td>
</tr>
</tbody>
</table>

TABLE II: Typical display characteristics for representative display classes. In Head-Mounted Displays, a lens between display and eye widens the covered FoV.
TABLE III: Bit rate savings of adaptive sampling frequency (ASF) over constant sampling frequency (CSF). Average bit rate \( r \) is given in megabits per second, frequencies in Hz. The bit rate savings in the rightmost two columns are the relative difference between the CSF bit rate and the ASF bit rate with the corresponding highest frequency \( f_{\text{max}} \) computed using Eq. (24).

defining \( f_s \) for CSF), and most of the time there is little to no motion (leading to a low \( f_c \)). On the other hand, with constant speed \( v \), there would be no difference between ASF and CSF. From these examples it becomes clear that ASF does never increase bit rate \( r \) compared to CSF if the video is played back at a frequency such that temporal sampling is imperceptible to humans.

VI. CONCLUSIONS

Our psychophysical study showed that the critical temporal video sampling rate depends on both object speed and camera exposure time. Based on the retrieved data, we built a model to compute this sampling rate as a function of speed as well as exposure time. In experiments, we have seen that playback at an adaptive sampling frequency defined by this model on average yields a 26% bit rate saving compared to constant sampling frequency playback.

This work will be used to refine the frame skipping algorithm in [2] and make temporal sampling imperceptible to viewers. However, the results of this paper are not constrained to that usage scenario, they can be applied to any video application with variable frame rate. For example, the frame rate of a video can be well below the perception threshold, but at a level that provides a user experience equal to a given, fixed frame rate video playback. This will reduce video bit rate, while not changing the perceived video quality.

Computer-generated graphics (CG) can also profit from our results: using a speed-dependent frame rate can reduce energy consumption in low-motion scenes and increase user satisfaction in motion-rich scenes. Furthermore, CG applications can model exposure time as in [26] to reduce the required frame rate.

REFERENCES

Christoph Bachhuber received the B.Sc. degree in electrical engineering and information technology from Technical University of Munich (TUM), Germany, in 2012 and the corresponding M.Sc. degree in 2014. He joined the Chair of Media Technology at TUM in 2014, where he is a member of the Research and Teaching Staff and is working towards his Ph.D. degree. His research interests include delay in video communication, vision-based control, augmented reality, and video coding.

Amit Bhardwaj received his B.Tech. degree in Electronics and Communication Engineering (ECE) in 2009 from YMCAIE Faridabad, Haryana, India, and his M.Eng. degree in ECE in 2011 from the Delhi College of Engineering, Delhi, India. In July 2011, he joined the department of Electrical Engineering (EE) at the Indian Institute of Technology (IIT) Bombay, Mumbai, India to pursue his Ph.D. degree, and completed the degree in June 2016. From June 2016 to January 2017, he was a research associate in the EE department at the IIT Bombay, Mumbai. From February 2017 to December 2017, he was an Alexander von Humboldt post-doctoral fellow at the Chair of Media Technology, Technical University of Munich, Germany. Since February 2018, he is a post-doctoral researcher at the Haptics and Virtual Reality (HVR) lab, Pohang University of Science and Technology (POSTECH), South Korea. He is a co-author of the book *Kinesthetic Perception - A Machine Learning Approach* published by Springer. His current research interests are in the area of data-driven haptic modeling and rendering, haptic perception, haptic data communication, signal processing, and applications of machine learning.

Rastin Pries studied Computer Science at the University of Wuerzburg (Germany). From 2004 to 2012 he was a member of the research staff of the Chair of Communication Networks at the University of Wuerzburg, where he received his doctoral degree in 2010. From 2012 to 2015 he was a technical consultant at VDI/VDE-IT for publicly funded research projects. In 2015 he joined NOKIA Bell Labs in Munich, Germany. His current research interests are the applicability of network slicing, NFV and data plane programmability concepts to 5G mobile networks as well as networked and interactive multimedia systems using edge clouds.

Eckehard Steinbach (M96-SM08-F15) studied Electrical Engineering at the University of Karlsruhe (Germany), the University of Essex (Great Britain), and ESIEE in Paris. From 1994 to 2000 he was a member of the research staff of the Image Communication Group at the University of Erlangen-Nuremberg (Germany), where he received the Engineering Doctorate in 1999. From February 2000 to December 2001 he was a Postdoctoral Fellow with the Information Systems Laboratory of Stanford University. In February 2002 he joined the Department of Electrical Engineering and Information Technology of Technical University of Munich, Germany, where he is currently a Full Professor for Media Technology. His current research interests are in the area of audio-visual-haptic information processing and communication as well as networked and interactive multimedia systems.