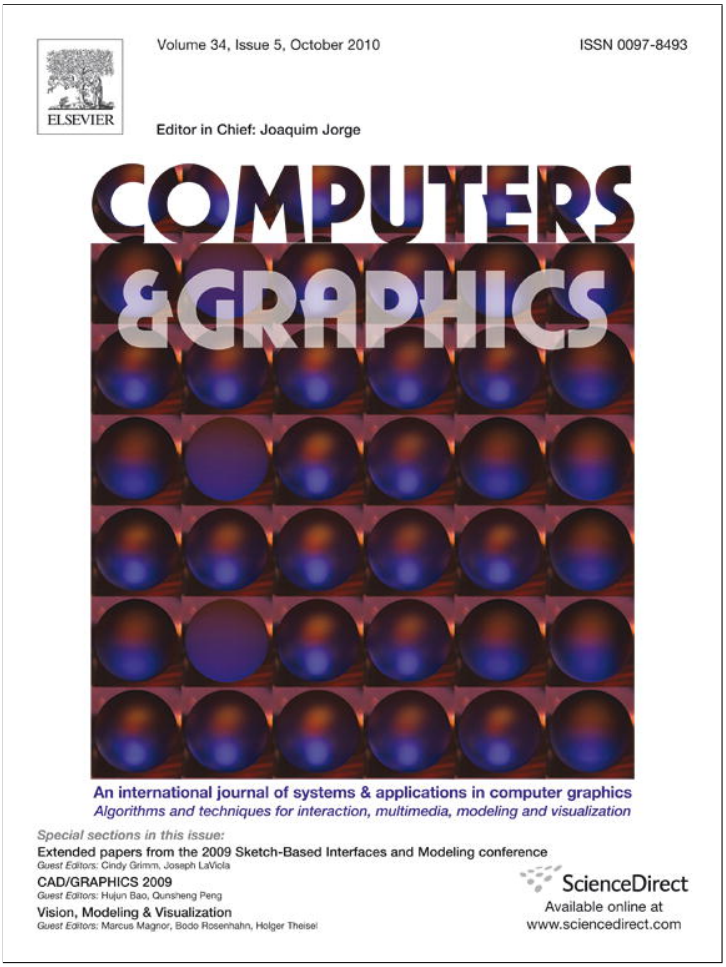

Computers & Graphics 34 (2010) 575–584

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Real-time temporal shaping of high-speed video streams

Martin Fuchs [a,*], Tongbo Chen [b], Oliver Wang [c], Ramesh Raskar [d], Hans-Peter Seidel [e], Hendrik P.A. Lensch [f]

[a] Princeton University, USA
[b] Rochester Institute of Technology, USA
[c] UC Santa Cruz, USA
[d] MIT Media Lab, USA
[e] MPI Informatik, Germany
[f] Ulm University, Germany

ARTICLE INFO

Keywords:
Computational videography
Real-time
Sampling
Temporal filtering

ABSTRACT

Digital movie cameras only perform a discrete sampling of real-world imagery. While spatial sampling effects are well studied in the literature, there has not been as much work in regards to temporal sampling. As cameras get faster and faster, the need for conventional frame-rate video that matches the abilities of human perception remains. In this article, we introduce a system with controlled temporal sampling behavior. It transforms a high fps input stream into a conventional speed output video in real-time.

We investigate the effect of different temporal sampling kernels and demonstrate that extended, overlapping kernels can mitigate aliasing artifacts. Furthermore, NPR effects, such as enhanced motion blur, can be achieved. By applying Fourier transforms in the temporal domain, we can also obtain novel tools for analyzing and visualizing time dependent effects.

We study the properties of both contemporary and idealized display devices and demonstrate the effect of different sampling kernels in creating enhanced movies and stills of fast motion.

1. Introduction

Today's cameras are able to capture scenes at frame rates far exceeding human visual requirements. These high-speed cameras have so far been used mainly for machine vision applications, but now gain prevalence as consumer devices. We feel that there is a new opportunity to use these cameras for computational videography. Specifically, we show that even if the output is intended for a human audience, we can exploit the captured high frequency signal to present improved videos at common frame rates (≈ 60 Hz).

Historically, video and film cameras have performed a rather simple temporal filtering: each single frame integrates the exposure of a different, non-overlapping time period. Depending on the shutter shape and its movement characteristics (as in rolling shutters), the recorded video will create a different viewing experience. The most prominent temporal artifacts are the wagon-wheel effect and non-continuous motion of fast moving objects. These effects can only be removed if temporal pre-filtering is applied prior to sampling the animation.

The ideal pre-filter kernel depends closely on the characteristics of the output device that will be used to display the video. Contemporary display technology, however, is very diverse in its construction, and accordingly, we cannot offer a single optimal pre-filter shape that works for every display type. In this article, in extension of its previous publication at the VMV 2009 conference, [1] we therefore instead discuss the temporal characteristics of different, typical output devices and the construction of an optimal sampling filter given the characteristics of the output device. We proceed to discuss the behavior of simple, abstract filter shapes and their effect on the videography temporal pipeline.

Making use of these filters, we present a novel computational imaging system which performs real-time temporal pre-filtering to dampen temporal aliasing. The system allows for temporally overlapping filters, which are a prerequisite for successful anti-aliasing. As the shape and extent of the temporal filter in our system can be chosen arbitrarily, we can perform different filtering operations, such as optimally pre-filtering for a given output kernel or artistically emphasizing or modulating motion blur.

Furthermore, we demonstrate specialized filter banks for analyzing the signal in the Fourier domain, in order to understand

* Corresponding author. Tel.: +1 609 258 0558; fax: +1 609 258 1771.
E-mail addresses: mfuchs@cs.princeton.edu (M. Fuchs), tongbo@cis.rit.edu (T. Chen), owang@soe.ucsc.edu (O. Wang), raskar@media.mit.edu (R. Raskar), hpseidel@mpi-inf.mpg.de (H.-P. Seidel), hendrik.lensch@uni-ulm.de (H.P. Lensch).
URL: http://www.cs.princeton.edu/~mfuchs/ (M. Fuchs).

0097-8493/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2010.05.017

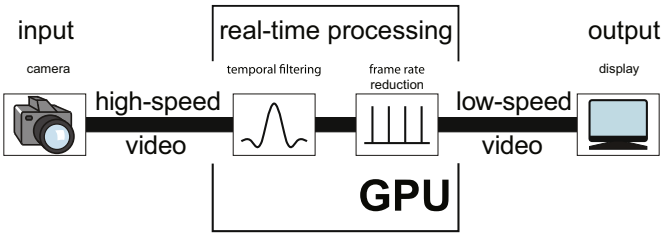

Fig. 1. Processing pipeline of the proposed camera system.

and enhance video content based on its temporal behavior, e.g. emphasizing or de-emphasizing motion.

Our system consists of a high speed camera coupled to a high performance GPU (Fig. 1). We demonstrate online recording and processing of 500 Hz input video at 1 MPixel mapped to a 60 Hz output. While the necessary compute power is currently provided by a GPU, FPGAs in consumer cameras are already close to being able to perform the proposed filtering inside a single device.

Our main contributions are:

- a discussion of temporal sampling with pre-filtering and reconstruction for videography (Sections 3 and 4),
- an evaluation of temporal behavior of contemporary display techniques (Section 5),
- a discussion of both optimal and common ad-hoc kernel sampling kernel types (Sections 6 and 7),
- a prototype system that performs real-time temporal filtering through super-sampling with overlapping kernels of arbitrary shapes (Section 8), and
- a Fourier camera application that can perform online image processing in the temporal Fourier domain (Section 9) such as frequency analysis and motion enhancement.

We demonstrate the following real-time applications: suppression of temporal aliasing in videos, non-photo realistic motion blur for videos and still images, and real-time Fourier-space processing.

2. Related work

Motion blur: In the context of rendering, temporal effects have been analyzed to remove aliasing by distributed sampling [2,3], to speed up the computation of animation sequences by frame-less rendering [4], and to faithfully create motion blur with photon mapping [5], to name a few. They can be synthesized using the accumulation buffer [6].

For video cameras, a simple way to create motion blur is to blend successive frames. However, this reconstruction kernel is not optimal. Brostow and Essa [7] proposed a method to add motion blur to stop motion animations by estimating the optical flow between the two images and then smearing the pixel colors along the trajectories. The analysis of optical flow and image alignment has been further used to correlate the acquired image samples over time, reducing noise in low-light conditions [8,9], and to estimate and extend the motion in a scene [10]. Hardware solutions for online spatio-temporal filtering for noise reduction using a spatio-temporal bilateral filter over a small window have been proposed as well. All these techniques are based on a regularly sampled video stream without altering the temporal filtering characteristic of the camera. The generated output significantly depends on the performance of the optical flow estimation. In our setup, we change the temporal filter kernel in order to produce the desired effect rather than relying on image-space vision algorithms.

Techniques for temporal filtering: The temporal filtering characteristics of a camera can be changed in a couple of different ways: One of the earliest controlled temporal filtering techniques made use of stroboscopes to create multi-exposure images of high-speed motion inside a single frame, (see Collins and Bruce [11]). In Section 8.3, we will demonstrate that we can obtain a similar effect with he help of an appropriately chosen filter kernel, without influencing the illumination of the scene in any way.

Recently, at a very small time scale, the photorefractive effect in photonic crystals has been used to implement a temporal high or low-pass filter for rather short time intervals [12,13]. Shechtman et al. [14] combined a set of video cameras to produce space-time super resolution videos which allowed for off-line temporal filtering. Bennett and McMillan [15] perform filtering on standard frame-rate video to create time-lapse output. They propose a virtual shutter for additional effects. We will present an online system that requires only a single high-speed camera.

Wilburn et al. [16] employed a multi-camera array, to compose an image where the length of the temporal filter can be chosen adaptively to the scene content. In our application we will process the video stream of a single high-speed camera to perform temporal filtering.

To fight motion blur, Raskar et al. [17] augmented a traditional camera with a high-speed ferro-electric LCD shutter. The time sequence of the shutter implements a broad band filter kernel that allows for reconstructing of sharp images of moving objects. This setup can in principle be used to shape the temporal filter in a fashion similar to strobe illumination in an on-and-off exposure sequence. However, it is inherently restricted to non-negative filter functions. In addition, as the integration will still be done within a single frame only, shaped overlapping filters are not possible.

Another interesting way to alter the spatio-temporal filtering is to move the camera, which Levin et al. demonstrated successfully for removing the effects of motion blur [18]. In their setup, the shape of the temporal filter through motion cannot be arbitrarily controlled.

Smart cameras: The design of our system relies on a tight coupling of the recording high speed camera and a high-performance compute platform. This design is rather similar to smart cameras for motion capturing, which record at a very high frame rate and then detect markers inside the camera [19–21]. However, they do not deliver a video stream as output but rather a compact representation of the marker trajectories. Smart cameras operating at standard video frame rates (e.g. [22,23]) offer real-time video manipulation but typically do not provide additional means to control the temporal filtering. Recently, the first consumer cameras appeared (such as the Casio Exilim Pro EX-F1) that provide high speed capture capabilities at rather low resolution. Specialized hardware compresses the video stream in real-time indicating that the necessary compute power for online temporal filtering within a consumer camera is within reach.

Temporal shaping for display devices: A considerable body of related work exists which specifically addresses the suppression of motion blur that is introduced in liquid crystal displays (LCDs) with long response times. Methods used to reduce the motion blur of LCDs can be classified according to three different goals: The first type endeavors to directly reduce the response time, for instance, by means of overdrive [24], optically compensated bend mode LCD [25] and dynamic capacitance compensation [26]. Another type aims at artificially increasing high frequency content through reducing the hold time, such as by including black data insertion [27], scanning back light [28], frame rate doubling [29], and gamma adjusted frame duplication [30]. Finally, a third type of work compensates for the luminance integration along the trajectories of moving objects, including motion adaptive edge compensation [31], edge enhancement, motion compensation

filtering [32,33], deconvolution based preprocessing [34] and a non-iterative filter bank technique [35].

In this article, we assume a high-fidelity, high-speed camera as recording device and focus on the processing of the incoming video stream, so as to optimize the input for the display, which we treat as a black box.

3. Temporal prefiltering

Let us first address the problem of avoiding temporal aliasing by optimal filtering in the temporal domain.

Due to recording with finite exposure times, every digital camera already performs some pre-filtering as part of the image capture. Conversely, every display device for time-variant data creates a time-continuous signal by means of a reconstruction filter. In this article, we take the properties of the monitor, including its sampling rate, as given, and investigate the choices for the camera's pre-filter depending on possible reconstruction filters on the monitor side. Before we discuss the relationship between these filters with the help of sampling theory, we will now summarize a mathematical model for image formation in a digital camera.

3.1. Image formation model

Consider a digital image I. The image value $I(x,y)$ at pixel position (x,y) corresponds to the amount of energy accumulated in the sensor over the exposure time. It can be expressed as an integration over time t of the flux $\Phi(x,y,t)$ arriving at that pixel, and a measurement kernel $m(t,x,y)$ encoding the temporally varying response:

$$I(x,y) = \int_{-\infty}^{\infty} \Phi(x,y,t) \cdot m(t,x,y)\, dt. \tag{1}$$

With mechanical shutters, there is always a non-trivial dependence of m on (x,y), as the shutter moves with a finite speed across the sensor. Unless used for artistic effect, these are undesired properties not present in many digital cameras that use an electrical shutter. We will therefore disregard the dependence of m on (x,y), and treat it as one-dimensional.

4. Imaging and sampling theory

Eq. (1) can in fact be interpreted as temporal sampling of a time-variant signal, so we can apply sampling theory considerations on its shape. As all pixels (x,y) are treated independently, we can focus on a single pixel, and call its time-variant signal $s(t)$. A digital movie camera is a device that generates a set of samples c_i at discrete points in time t_i, so that

$$c_i = \int_{-\infty}^{\infty} s(t) \cdot m(t_i - t)\, dt. \tag{2}$$

Correspondingly, a monitor or digital display device takes the discrete pixel sampling c_i, and generates a continuously defined output approximation

$$\tilde{s}(t) = \sum_i c_i \cdot r(t - t_i) \tag{3}$$

of the input signal with a reconstruction kernel $r(t)$.

We know from the work of Shannon [36], that, should $s(t)$ be band-limited with a frequency of $\frac{1}{2}\nu$, i.e., the signal does not contain any energy in any higher frequency band, it can be completely represented by sampling it with a rate of ν, yielding a discrete representation $(c_i)_{i \in \mathbb{Z}}$ from which a perfect reconstruction is possible. Shannon's observations tell us that a perfect reconstruction

$$\begin{aligned} s(t) &= \sum_i c_i \cdot r(t - t_i) \\ &= \sum_i \int_{-\infty}^{\infty} s(t') \cdot m(t_i - t')\, dt' \cdot r(t - t_i) \end{aligned} \tag{4}$$

is possible for the choice of

$$r(t) = m(t) = \mathrm{sinc}(t \cdot \nu) \quad \text{for } \mathrm{sinc}(x) = \frac{\sin(x)}{\pi x}. \tag{5}$$

Prefiltering by convolution with $\mathrm{sinc}(t)$ of the appropriate sampling frequency (multiplication with a box function in Fourier space) effectively removes all frequencies beyond the Nyquist limit. Shannon's theorem guarantees that the filtered signal can be reconstructed from the sampled sequence, but when capturing motion, almost arbitrary frequencies can occur in single pixels due to occlusions and dis-occlusions. If they are strictly filtered out, the output will contain ringing artifacts. Even worse, the sinc kernel has infinite support: even if the frequencies were limited, we would have to integrate over the entire video.

Meanwhile, as sampling theory has progressed [37], the relationship between $m(t)$ and $r(t)$ is much better understood. Overall, we want to approximate $s(t)$ as close as possible with some function $\tilde{s}$ (see Eq. (3)). In the least squares sense, this is a projection into the function space spanned by the $(r_i)_{i \in \mathbb{Z}}$,

$$r_i(t) \coloneqq r(t - t_i). \tag{6}$$

Unser [37] describes techniques to compute the kernel $m(t)$ for a given $r(t)$ which yields the best approximation $\tilde{s}(t)$ for the original signal $s(t)$. He also discusses the theoretical relationships in far greater depth than would be appropriate in this article.

We observe that the shape of the display reconstruction kernel $r(t)$ is crucial for a smooth reconstruction; in cases where it contains frequencies above $\nu/2$, reconstructing with $r(t)$ will introduce spurious frequencies into $\tilde{s}$ that in general no pre-filtering on the input-signal can prevent. These spurious frequencies cause so-called *aliasing artifacts* in the reconstruction.

5. Properties of real-world display technology

In order to achieve an optimal reconstruction for an output device by means of pre-filtering through supersampling, its temporal behavior must first be established. In this section, we investigate the temporal response of several displays which are representative of technologies widely available today.

LCD displays: In a first experiment, we investigated the temporal variation of a Dell 2005FPW flat panel monitor. We fed the display with a repeating pattern of four frames fully black, four frames fully white at 60 Hz, and observed it in a 1200 Hz video recorded with a Casio Exilim EX-F1 consumer camera. Fig. 2(a) shows the temporal behavior switching from black to white and back. As can be seen, the display needs more than one frame worth of time to reach a fully white state from a black state, but switches to the black state faster (up to camera precision).

In a second experiment, we used the same hardware configuration, but had the display switch between two mid-gray levels. Fig. 2(a) shows a scaled plot in comparison to the black–white–black experiment; it illustrates that the display actually needs more time to reach its illuminated state. From both observations, we conclude that this particular display cannot be accurately modeled as a linear system.

We have also investigated the behavior of a newer, more modern 24 in panel (Dell 2408WFPb). In order to increase accuracy, we recorded its response with a Basler A504kc digital camera in a 120 × 120 sub-frame of the sensor at 4000 Hz. While

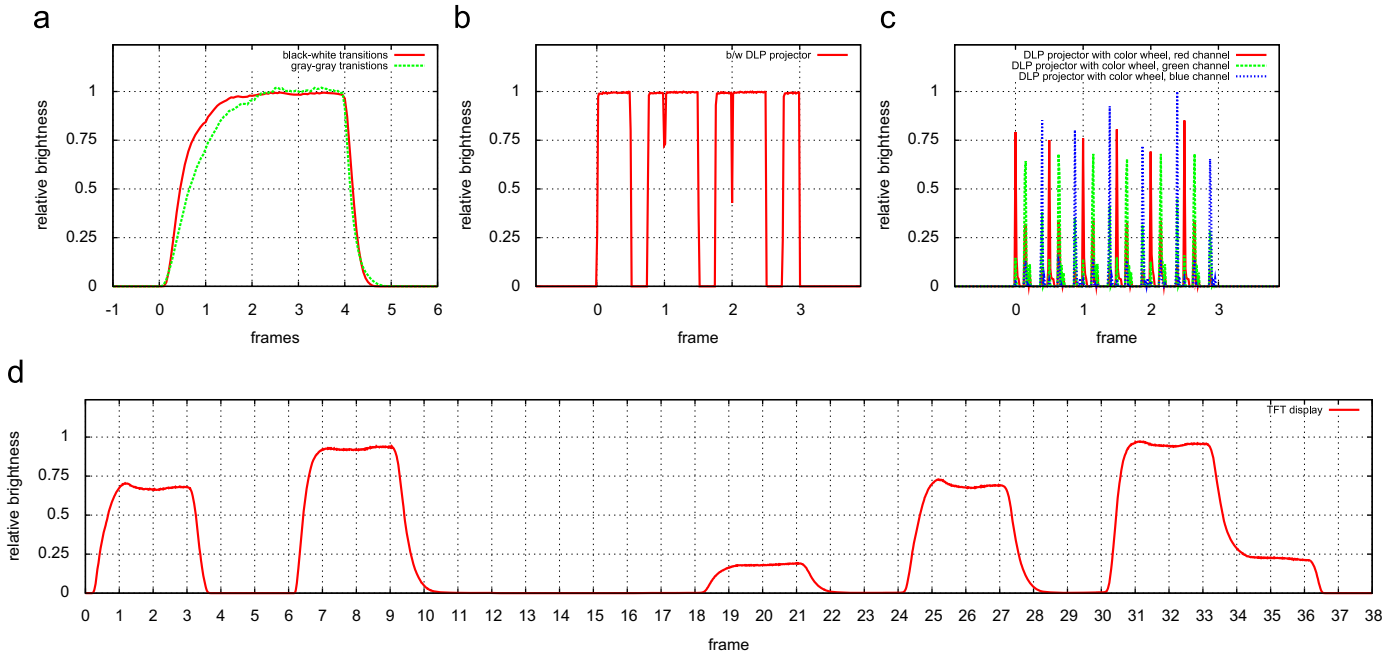

Fig. 2. Temporal response of real-world displays. (a) Common TFT displays expose a response characteristic that may extend into the next frame, the extent varying with for different gray level states. (b) DLP projectors dither temporally, and therefore behave strongly nonlinearly. (c) DLP Projectors with color filter wheel have high frequent behavior that cannot be observed precisely even with a 4000 Hz video stream. (d) A more modern TFT display switches more precisely. (a) Dell 2005FPW TFT monitor: black–white vs. gray–gray transitions; (b) black-and-white DLP projector displaying constant gray; (c) DLP projector with color wheel displaying constant gray; and (d) Dell 2408WFPb TFT monitor going through transitions of different gray levels.

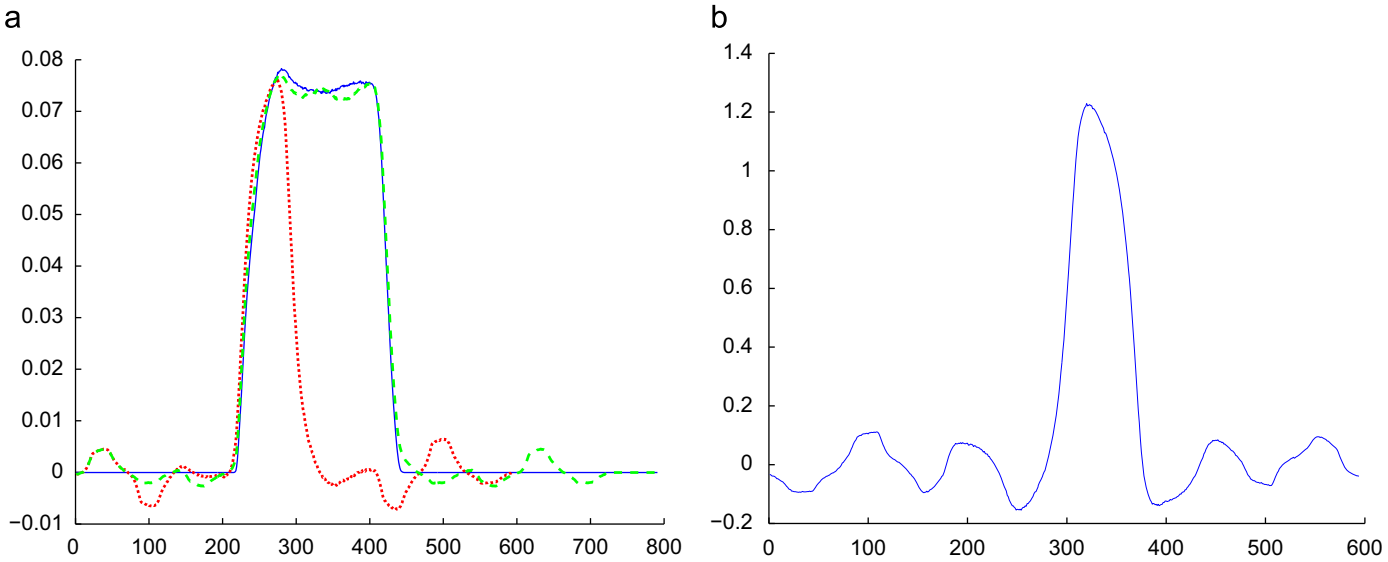

Fig. 3. Approximation of the temporal switching behavior of an LCD display in a linear system. (a) Blue/solid: Temporal response switching from black to 75% gray for three frames, then switching back). Red/dotted: A least-square optimal estimation of the display's reconstruction kernel. Green/dashed: appearance of the input signal assuming the reconstruction kernel. It is close to the input signal, but suffers from ringing artifacts. (b) Optimal pre-filter kernel for the reconstruction kernel.

this configuration may lose some precision for the darkest display states due to the low exposure time and the constant read-out noise, we can obtain reasonably accurate estimates of the display behaviors in the bright states.

Fig. 2(d) shows a plot of the average display brightness while it transitions through different gray states, which are kept constant for three frames each at 60 Hz input frame rate. As can be seen, this display still shows slightly different behavior for different combinations of states, overall, though, the different state changes are qualitatively close.

Therefore, we have chosen the 0%–75%–0% part of this sequence, which for an assumed gamma value of 2.2 corresponds roughly to a 0%–50%–0% brightness distribution, as representative of its global behavior, and proceeded to estimate its reconstruction function $r(t)$ based on the known input of three constant pulses padded by zeros.

While the computation of this kernel is a deconvolution in nature, we found it more stable and effective to derive it as a direct least-squares solution of a linear equation system that searches for the function that, when shifted three times by one

frame width, approximates the input curve best. Fig. 3(a) shows the result of this evaluation: the measured input curve is drawn in blue and the estimated reconstruction kernel $r(t)$ in red. Convolving the input pulses with $r(t)$, we obtain the green curve, which represents the predicted behavior of the display when modeled with the newly found kernel. At the flanks of the original signal, it produces a very good match. However, outside the flanks, noticeable ringing occurs.

Using the method discussed by Unser et al. (37), we can estimate the optimal pre-filter kernel for this display, shown in Fig. 3(b). It also exposes noticeable ringing, and contains negative coefficients. Because of that, it cannot be used precisely for pre-filtering, as negative values in the sampled sequence could occur that cannot be communicated to a monitor with one of the established display protocol standards.

Digital mirror device-based displays: While the previous observations can be expected to apply qualitatively to LCD-based projection systems as well, the situation is fundamentally different for displays based on micro-mirror arrays because of the different image formation process.

Fig. 2(b) and (c) shows the temporal response of DLP projectors that display a signal that switches from black to a mid-gray value for three frames, and then again to black.

In the black-and-white projector displayed in (b), the outer flanks of the signal are very steep, almost perfectly rectangular. However, as the display technology is inherently restricted to displaying binary signals, the projector dithers its output, which is evident in the plot shown. This illustrates that its response is highly nonlinear; modeling its effect may, however, be possible if the properties of the human visual system are taken into account.

For a color DLP projector, as shown in Fig. 2(c), the response is even more difficult to model, as the different colors are actually interleaved; they change their state so quickly that even with the high speed recording technique we employed, an accurate measurement is not possible.

6. Temporal characteristics of idealized displays

As we have seen in the previous section, existing display technology differences have a large impact on the temporal behavior, rendering a universal, yet precise filtering approach virtually impossible. Nevertheless, we can muse on idealized displays for which perfect temporal shaping of the input signal is possible, and discuss the impact of design choices on their characteristics.

In this section, therefore, we will now discuss typical reconstruction functions $r(t)$ of idealized output devices with increasing smoothness. Specifically, we will look at the B-spline basis functions, and the accordingly least-squares optimal kernel shape $m(t)$ for pre-filtering. We perform an experiment and provide a synthetic scene with a spinning five-pointed star (see Fig. 4). We discretize each sampling interval $T=1/\nu=1$ into 32 steps for simulation purposes. For each of the choices of $r(t)$, sampling is performed with the corresponding L_2-optimal pre-filtering kernel $m(t)$. The supplemental video shows the results; we invite the reader to determine which one suits the reconstruction by his/her monitor best.

The selected output filters mimic different conceivable behaviors:

Box function (zero-order-hold) with width T and height 1, i.e. $r(t)=\mathrm{rect}(t)$. The discretized reconstruction filter r_τ forms an orthonormal system, and the L_2-optimal pre-filtering $m(t)$ and $r(t)$ actually are the same functions.

Triangle function with width $2\cdot T$ and height 1, i.e. $r(t)=\mathrm{tri}(t)=\mathrm{rect}(t)*\mathrm{rect}(t)$. This implies that the output device performs a linear interpolation of the sample values. The reconstruction suggests a more continuous rotation; however, the appearance of the frames differs strongly ($T=0$ vs. $T=\frac{1}{2}$). Due to the multitude of peaks in the pre-filter $m(t)$, ringing becomes apparent.

Higher-order B-splines, i.e. $r(t)=\beta^i(t), i\geq 2$. For the higher-order B-spline basis functions, obtained by convolving $\mathrm{rect}(t)$ i times with itself, we observe that the consistency between in-between and sampled frames increases. At the same time, the broad support of these kernels largely reduces the contrast between foreground and background.

Sinc: $r(t)=\mathrm{sinc}(t)$ is the optimal pre-filter as it suppresses frequencies beyond half the sampling limit most effectively. In the simulation, the sinc is windowed to an interval $[-12.5, 12.5]$. The reconstruction shows overly dark areas for $T=\frac{1}{2}$ where the reconstructed signal was actually negative due to ringing. As in the $\mathrm{rect}(t)$ case, the r_τ form again an orthonormal system, and $m(t)$ and $r(t)$ coincide.

Transient: In most physical systems a transition from one state to another follows an exponential function, such as it might occur if alternating current is dampened by a capacitor. The filter

$$r(t)=\begin{cases} e^{-\lambda t} & \text{if } 0\leq t<1,\\ 1-e^{-\lambda(t+1)} & \text{if } -1\leq t<0,\\ 0 & \text{otherwise,}\end{cases} \tag{7}$$

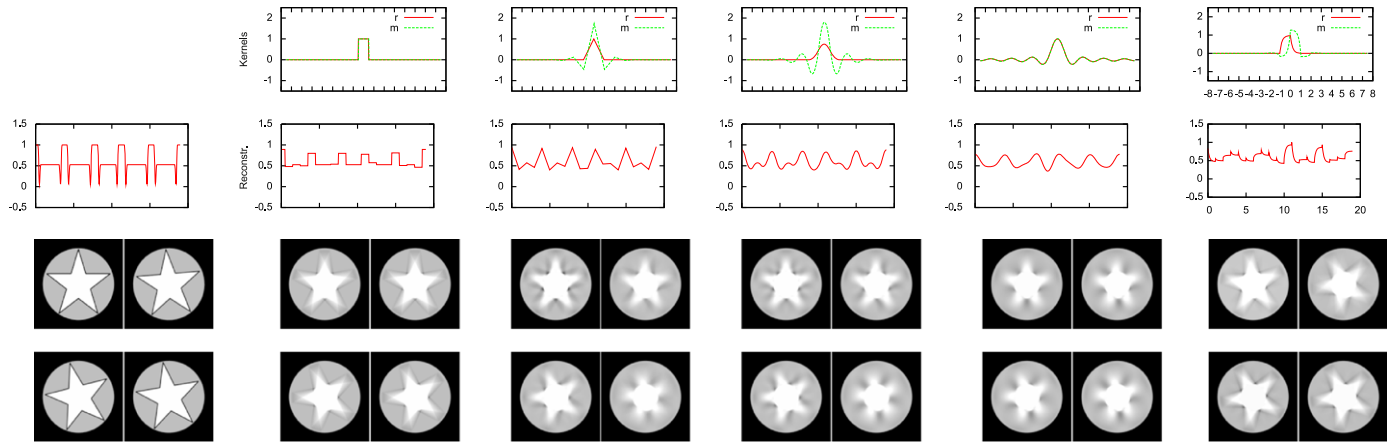

Fig. 4. Filtering results for a spinning star. From top to down: shape of the pre-filter and the reconstruction kernel, intensity profile for a single pixel, and reconstructed images for individual time steps ($t\in 0;\frac{1}{4};\frac{1}{2};\frac{3}{4}$ clockwise starting top left) within one frame.

describes another possible time dependence of a digital monitor, simulated in our case with $\lambda = 5$. It stands out from the other discussed functions as being asymmetric; it further has the remarkable property that it yields an in-between image for $T = \frac{3}{4}$ which is sharper than the images at the sampled positions for $T \in \{0,1\}$.

6.1. Common observations

In the examples above, the optimal pre-filter for all but the box kernel extend over a period of two samples or more. Following Shannon's argument, a camera implementing any reasonable pre-filter kernel *needs* to accumulate the data of several frames worth of exposure into a single output frame. This can in the general case be realized computationally, but for most kernels it requires the same sub-exposure to be counted towards several distinct frames with different weights.

The pre-filtering in most digital cameras resembles a rectangular kernel $m(t)=\mathrm{rect}(t/w)$ with width w. Assuming that the kernel spans the entire duration between two frames ($w=1$), and assuming an unrealistic output device with $r(t) = \mathrm{rect}(t)$, this sampling kernel is L_2-optimal. Most often, though, the exposure time is much shorter than the frame duration to avoid saturation, and the system produces severe aliasing effects such as the "wagon-wheel moving backwards" illusion or jagged, discontinuous motion (see supplemental video).

Another aliasing effect that is quite apparent in all filters with extended support is ringing. The ringing is an effect of a windowed filter kernel in the Fourier domain that manifests itself as an overshooting signal in the spatial domain. In our example, it is due to the unbounded frequency of the input. These high frequencies are not necessarily sufficiently suppressed by the L_2-optimal pre-filters.

7. Evaluation of further kernel types

In the following paragraphs, we will discuss a selection of further, simple kernel shapes without a direct connection to sampling theory and their effects. This time, we choose kernels that are free from negative values, and therefore are ideally suited to be used on video streams to be sent to contemporary displays through standardized video connectors that only support non-negative values.

Unlike in the previous discussion we will not take the reconstruction into account and only perform a shaped exposure filtering for visually pleasing results. Fig. 5 shows the results for two successive frames of the rotating star sequence; the entire sequence is visualized in the supplemental video. The selected filters provide relatively little ringing.

The kernel types *point*, *halfbox* and *fullbox* correspond to point sampling and rectangular kernels of width $w = T/2$ and $w = T$, respectively. They represent the results achievable with a traditional camera. While the convolution for $w = T$ produces at the least some overlap between successive frames, the first two cases skip some in-between sub-frames completely, yielding a stuttering, jagged appearance in the video.

The *triangle* and *Gaussian* kernels, perceptually close in appearance, provide a smooth, continuous transition between frames. Each individual output frame, however, looks rather smooth, though.

In their work on reconstruction filters [38], *Mitchell and Netravali* proposed a class of piece-wise cubic filters for reconstructing point sampled data and demonstrated their effectiveness with a user study. The results of using these filters, for parameters $(B,C) = (\frac{1}{3},\frac{1}{3})$ and $(B,C) = (\frac{3}{2},-\frac{1}{4})$, respectively, yield also smooth transitions between dark and bright pixels, but seem less fuzzy than the Gaussian and the triangle filters.

For a non-photorealistic (*NPR*) effect, we simply took a nonlinear exponential function $m'(t) = e^{(-1+t)2}$, blurring it slightly. It highlights a sharp exposure, but pulls a trail of continuously falling pixel values behind, akin to the afterglow of an exponential decay process. In order to emphasize the effect, we have extended the filter width to four frames.

Finally, the *strobe* kernel simulates an illumination with a stroboscope. The stills of different positions in time add up to an overlay image. In contrast to a stroboscope illumination, though, we can achieve the effect without influencing the scene illumination solely by choosing an appropriate filter kernel.

These example kernels demonstrate a range of possible temporal filtering characteristics. The best choice is dependent of the desired effect. In the following section, we will study their behavior in a real-world scenario.

8. Real-time processing system

Having discussed the theoretical and practical aspects of temporal behavior with synthetic analysis, we will now introduce the design aspects of our hardware and software prototype.

8.1. Hardware configuration

In order to approximate arbitrarily shaped filters on a continuous signal we sample at a much higher frame rate than the final display. Using a Basler A504kc camera, we capture at 480 Hz at a resolution of 1000×1024 or 500 Hz at 240×256 for the Fourier analysis in the next section. We stream the captured frames to an NVIDIA GeForce GTX 280 graphics card. The temporal filtering is implemented in Cuda.

The system is capable of performing the necessary processing for the high speed video stream in real-time, continuously generating an output video at 60 Hz. As an alternative capturing device, we employ a Casio Exilim Pro EX-F1 camera to record high speed video at 300 Hz.

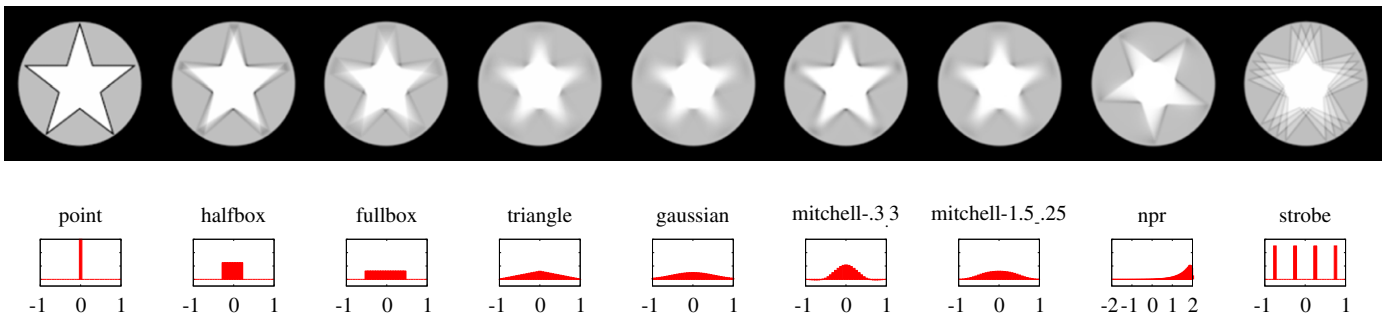

Fig. 5. Excerpt from a simulated sequence of a spinning star and the generating kernels (bottom).

8.2. Temporal filtering pipeline

The incoming frames are transformed into a smaller number of output frames. Each output frame is obtained by convolving the corresponding input frames with the filter kernel. As the filter kernels for adjacent frames can overlap, input frames can contribute to more than one output frames.

In order to minimize the required bandwidth we stream color-filter-array images into GPU memory and perform all operations on these images. Only for display, we run demosaicing by bi-linear interpolation and subtract the black frame at 60 Hz.

Given a specific frame rate reduction, the maximum filter length is only bounded by the processing speed and local bandwidth on the GPU.

8.3. Results for rigid body motion

Fig. 6 shows the effect of different filtering kernels on the repeating motion of a rotating fan. Using a one-point sample results in a jaggy motion with a strong wagon-wheel effect when the fan spins up or down. This aliasing effect is still present for the box filter but removed in the Mitchell–Netravali filter.

8.4. Results for stochastic processes

In Fig. 7 we visualize the effect of temporal filtering on stochastic motion with repeating patterns. Point sampling freezes the motion in time and renders rather sharp images. Note how the water stream is composed of individual droplets. At the same time the still frame hardly conveys the associated motion any more. In the output video, point sampling leads to the appearance of a rather random sampling. Using a box instead, all droplets are smeared into streaks, but the sequence still contains too high frequencies to render the sequence attractively. The Mitchell–Netravali (MN) filter on the other hand is too smooth. The vividness of the water and the flames is significantly dampened.

In our NPR filter we combined the spatial detail—however, slightly filtered—with the motion direction information close to

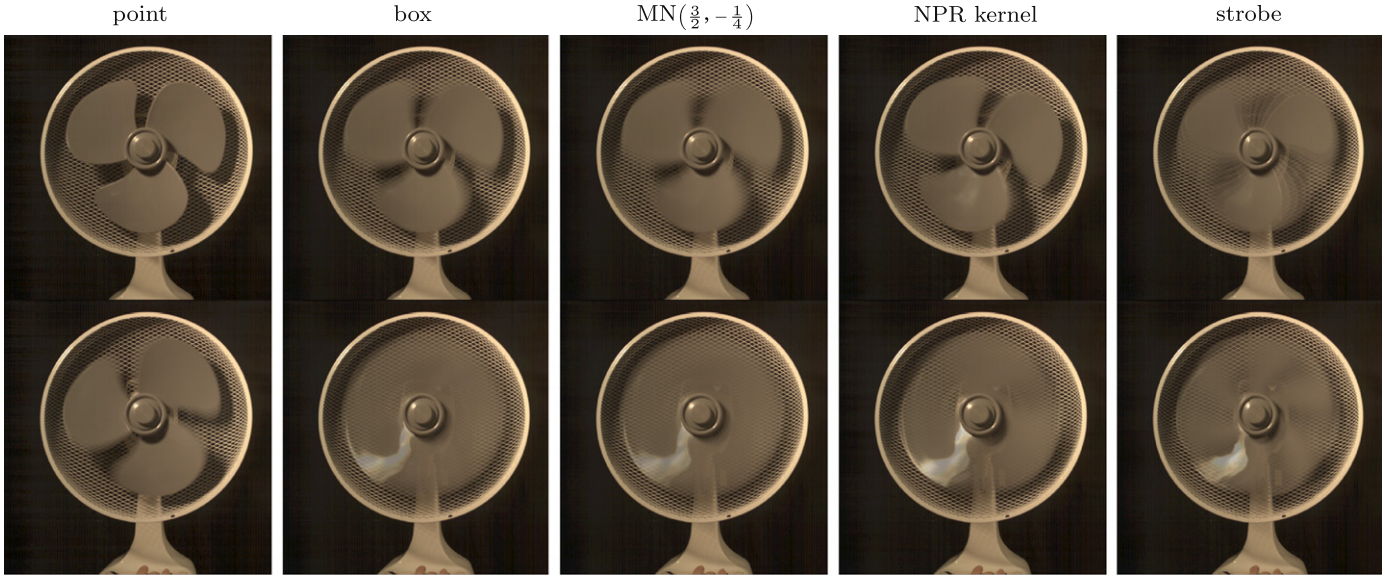

Fig. 6. Various filters applied to the video of a spinning fan at two different velocities. Aliasing effects are visible in the accompanying video for all filters except for the MN filter.

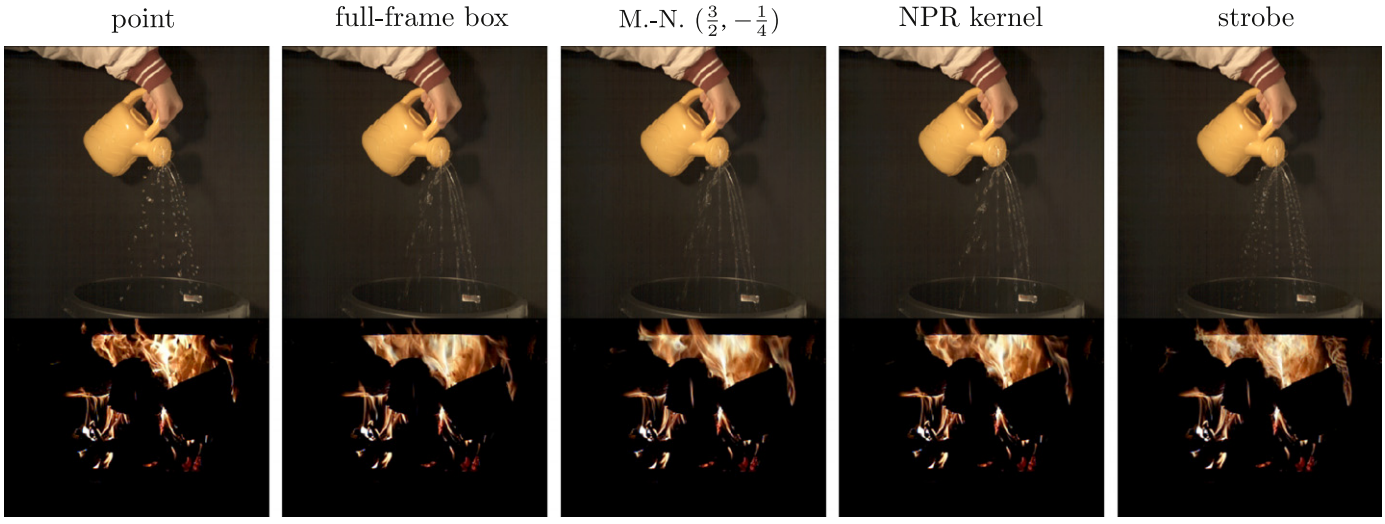

Fig. 7. Filtering results for stochastic processes. Depending on the applied filter individual particles or the motion direction is visible. The nonlinearly increasing filter combines the details of the point filter while indicating the motion direction.

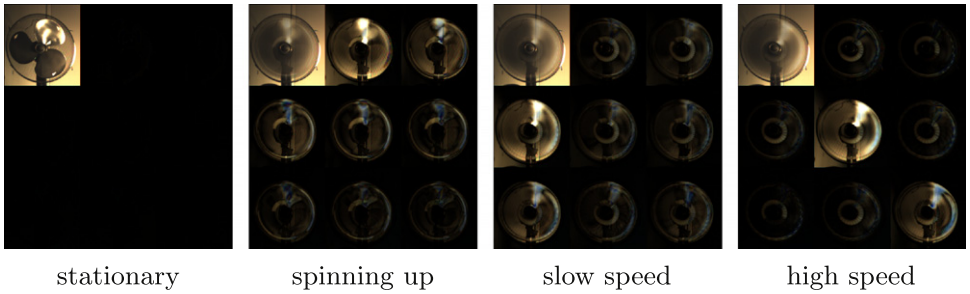

Fig. 8. Screen shots of the online temporal Fourier transform. Each window shows the first nine bands of the power spectrum, with increasing frequency from top left to bottom right. As the fan accelerates, the energy moves to higher bands.

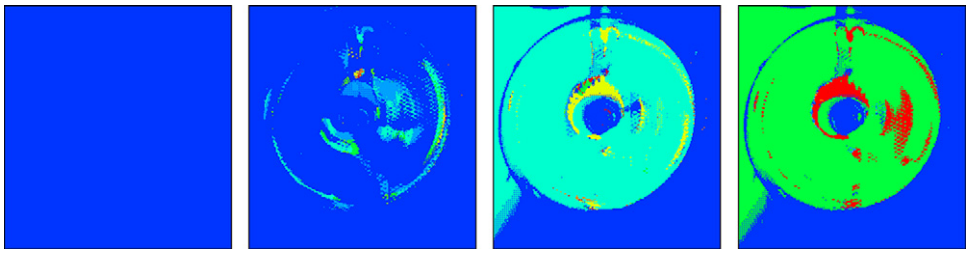

Fig. 9. Brightness-coded maximum frequency of the fan of Fig. 8 spinning at four different velocities.

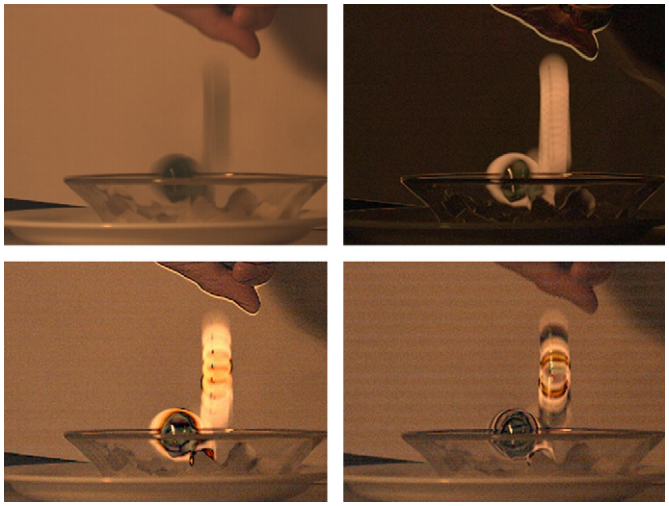

Fig. 10. Modified appearance of a video sequence in Fourier space: standard Gaussian filter (top left), absolute values after removing the DC band (top right), boosting low frequencies (bottom left), and boosting high frequencies (bottom right).

the MN filter. We argue that in the stills, this NPR filter summarizes the characteristics of the two stochastic effects better than any other filter. In addition, the video is crisp but far less random compared to the point sampling.

In the fire sequence, the strobe filter nicely shows the propagation of the reaction surfaces over time. While for reflecting objects this can be obtained using a strobe illumination, we can visualize this effect in real-time even for self-emitting media.

9. Fourier camera

The ability to perform the convolution of high speed video material in real-time makes it possible to visualize periodic or non-periodic movement in a novel way: by performing a discrete per-pixel temporal Fourier transform on the input. Per-pixel

Fig. 11. In contrast to a plain Gaussian filter kernel (left), boosting frequency components emphasizes the motion.

oscillations can be observed in different frequency-bands, with the zero-band displaying the temporal average, and higher bands showing the oscillations at different frequencies.

Implementation: The frequency analysis is performed by computing a sliding discrete Fourier transform over 32 frames captured at 500 Hz. We compute nine bands with a resolution of 240×256 each at 50 Hz. A different Fourier transform is calculated for each frame.

9.1. Fourier domain applications

Fig. 8 shows a screen capture of the power spectrum of a real-time decomposition of a spinning fan sequence into separate frequency bands. The fan rotates at three different velocities, thus generating clearly distinct distributions across the bands. By analyzing this data, we can perform frequency-based segmentation quite easily (Fig. 9). Note, however, that the result of the Fourier transform will be influenced by both the actual motion as well as the texture of the moving object.

As the computational performance of the proposed system is high, we can also compute the inverse Fourier transform, which makes editing in Fourier space possible. One possible application is the selective emphasis of some frequencies, as illustrated in Fig. 10.

In comparison to a standard Gaussian filter kernel, simple edits reveal movement structures: if the zero-order (DC) band is removed from the reconstruction, only moving scene parts can be seen, a selective frequencies boost triggers a motion trail effect. Selecting the frequencies influences spatial extent and visual contrast.

Fig. 11 shows the frequency boosting effect on thin plant leaves moving in wind. Here, the perceived motion is enhanced by the frequency edit—without analyzing or even tracking the motion behavior of individual scene components. The effect is similar to previous work [39] in introducing subtle ringing that creates a perceived motion effect.

Such simple, multiplicative edits correspond to appropriately chosen convolutions in the primal domain, where there is also more flexibility on the choice of the reconstructing kernel, as put forth in Section 3. However, these operations have more intuitive control in the Fourier domain; they may be understood as a video signal equivalent of an equalizer (EQ) circuit, which is a staple component of acoustic signal processing.

10. Limitations

Performing temporal pre-filtering by starting from a super-sampled sequence comes at a cost. As each sub-frame is exposed for a very short period (<2 ms), the number of recorded photons is limited. The signal to noise ratio is weaker compared to a single exposure for the entire frame. On the other hand, techniques such as [8,9] as well as our own footage successfully show that after integration of the recorded frame this effect is greatly reduced.

A limiting factor of our system is the available bandwidth both when transferring the camera data to the GPU which we only managed at 1000×1024 at 500 Hz compared to the maximum resolution of the camera (1280×1024) and on the GPU when integrating into multiple output frames.

11. Conclusions

In this article, we have discussed the interactions between recording and display devices in the temporal domain, and presented a computational videography system that allows for freely controlling the shape of the temporal filter that is applied when recording an animation. The system exploits the capabilities of a high speed camera augmented with sufficient compute power, a configuration that can be expected to become available in consumer cameras in the near future.

For optimal results, this temporal shaping of the video signal should take the characteristics of the output device into account. With the wide diversity of technology available today, this cannot be achieved in a uniform way. Nevertheless, standardized definition of display behavior has seen much progress in recent years, and can be observed even in consumer products containing circuitry enabling conformity to color standards as well as photometric display response.

We argue that, in order to prevent severe temporal aliasing, it is in any case necessary to perform an integration over the duration of several output frames, which requires to accumulate each incoming frame to more than one output frame, weighted by its relative position in the filter kernel, which becomes possible in the system we have presented in this article.

In the accompanying video we demonstrated the effectiveness of the proposed temporal filters, producing artifact reduced videos as well as more expressive stills for stochastic motion events such as water falls or flames.

Some of the filters could conceivably be approximated using a video camera operating at standard frame rates and a flash, the intensity of the latter being modulated over the duration of a frame. Our setup, however, does not necessitate casting any additional light into the scene, and supports overlapping integration periods natively. In addition, it is flexible enough to perform even more complicated processing on the input frames in real-time, such as the simulated motion trails from [15], shown in Fig. 12.

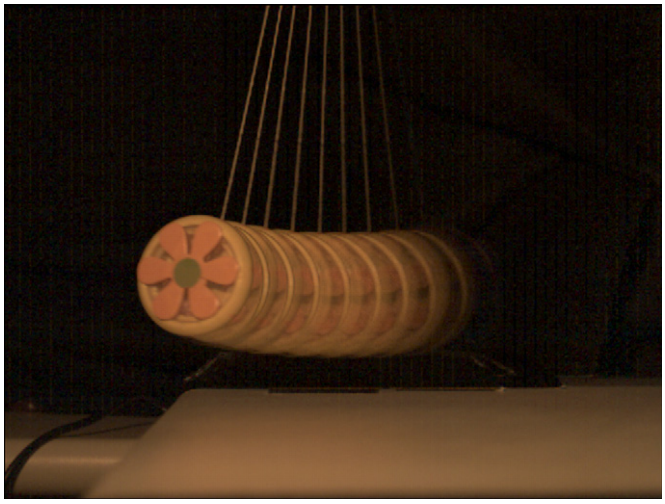

Fig. 12. A single frame of a video enhanced in real-time with motion trails.

As our system allows to freely change and control the shape of the temporal filter during online recording, it cannot only be optimized for the target display characteristics, but can also contribute an additional means of artistic expression. Thus, it introduces a temporal equivalent of the bokeh—the appearance change in photography caused by the choice of the aperture shape—to videography.

Acknowledgments

This work is an extension of a presentation at VMV 2009 [1] which was in large parts performed while Martin Fuchs, Tongbo Chen, and Oliver Wang were affiliated with the MPI Informatik. It has been partially funded by the DFG Emmy Noether fellowship (Le 1341/1-1) and the Max Planck Center for Visual Computing and Communication (BMBF-FKZ01IMC01).

The authors would like to thank Christian Fuchs from Ulm University for generously providing raw data of several of the display experiments, and to Connelly Barnes and Szymon M. Rusinkiewicz from Princeton University for many fruitful discussions.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version of 10.1016/j.cag.2010.05.017.

References

[1] Fuchs M, Chen T, Wang O, Raskar R, Seidel H-P, Hensch HPA. A shaped temporal filter camera. In: Proceedings of the vision, modeling, and visualization workshop 2009, Braunschweig, Germany, 2009, p. 177–86.
[2] Cook RL, Porter T, Carpenter L. Distributed ray tracing; 1998. p. 77–85.
[3] Potmesil M, Chakravarty I. Modeling motion blur in computer-generated images. SIGGRAPH Comput Graphics 1983;17(3):389–99. doi:〈http://doi.acm.org/10.1145/964967.801169〉.

[4] Bishop G, Fuchs H, McMillan L, Zagier EJS. Frameless rendering: double buffering considered harmful. In: SIGGRAPH '94: proceedings of the 21st annual conference on computer graphics and interactive techniques, 1994, p. 175–6 〈http://doi.acm.org/10.1145/192161.192195〉.
[5] Cammarano M, Jensen HW. Time dependent photon mapping. In: EGRW '02: proceedings of the 13th Eurographics workshop on rendering, 2002. p. 135–44.
[6] Haeberli P, Akeley K. The accumulation buffer: hardware support for high-quality rendering. In: SIGGRAPH '90: proceedings of the 17th annual conference on computer graphics and interactive techniques, 1990. p. 309–18. doi: 〈http://doi.acm.org/10.1145/97879.97913〉.
[7] Brostow GJ, Essa I, Image-based motion blur for stop motion animation. In: SIGGRAPH '01: proceedings of the 28th annual conference on computer graphics and interactive techniques, 2001. p. 561–6.
[8] Bennett EP, McMillan L. Video enhancement using per-pixel virtual exposures. In: ACM transactions on graphics (Proceedings of the SIGGRAPH 2005), vol. 24(3), 2005. p. 845–52.
[9] Telleen J, Sullivan A, Yee J, Wang O, Gunawardane P, Collins I, et al. Synthetic shutter speed imaging. Comput Graphics Forum 2007;26(3):591–8.
[10] Liu C, Torralba A, Freeman WT, Durand F, Adelson EH. Motion magnification. In: ACM transactions on graphics (Proceedings of the SIGGRAPH 2005), 2005. p. 519–26.
[11] Collins D, Bruce RR, editors. Seeing the unseen: Dr. Harold E. Edgerton and the wonders of Strobe Alley. MIT Press; 1994.
[12] Gao H, Zhang J, Yoshikado S, Aruga T. Photorefractive low-pass temporal filter. Opt. Commun 2002;203(3–6):363–9.
[13] Yang D, Kumar S, Wang H. Temporal filtering using time lenses for optical transmission systems. Opt Commun 2008;281(2):238–47.
[14] Shechtman E, Caspi Y, Irani M. Increasing space–time resolution in video. In: ECCV '02: proceedings of the seventh European conference on computer vision—part I, 2002. p. 753–68.
[15] Bennett EP, McMillan L. Computational time-lapse video. In: ACM transactions on graphics (Proceedings of the SIGGRAPH 2007), vol. 26(3), 2007. p. 102, doi: 〈http://doi.acm.org/10.1145/1276377.1276505〉.
[16] Wilburn B, Joshi N, Vaish V, Talvala E-V, Antunez E, Barth A, et al. High performance imaging using large camera arrays. ACM Trans Graphics 2005;24(3):765–76.
[17] Raskar R, Agrawal A, Tumblin J. Coded exposure photography: motion deblurring using fluttered shutter. In: ACM transactions on graphics (Proceedings of the SIGGRAPH 2006), vol. 25(3), 2006. p. 795–804, doi: 〈http://doi.acm.org/10.1145/1141911.1141957〉.
[18] Levin A, Sand P, Cho TS, Durand F, Freeman WT. Motion-invariant photography. In: ACM transactions on graphics (Proceedings of the SIGGRAPH 2008), vol. 27(3), 2008. p. 711–19.
[19] Vicon Peak, Camera MX 40 〈http://www.vicon.com/products/mx40.html〉; 2006.
[20] Motion Analysis Corporation, Hawk-I digital system; 2006.
[21] Optotrak, NDI Optotrak Certus Spatial Measurement 〈http://www.ndigital.com/certus.php〉; 2007.
[22] FastVision, FastCamera40 〈http://www.fast-vision.com/cameras/camera40.html〉; 2008.
[23] Vision Components, VC4472; 2008.
[24] Okumura H, Fujiwara H. A new low-image-lag drive method for large size LCTVs. J SID 1993;1(3):335–9.
[25] Hirofumi Wakemoto KN, Takimoto A. Advances in OCB mode LCDs improvement of moving picture quality and control of bend alignment. In: Proceedings of SPIE, vol. 6135, 2006. p. 613501–7.
[26] Lee S-W, Kwon S, Kim M, Souk J, Kim SS. Improved technology for motion artifact elimination in LCD monitors advanced DCC. Digest SID 2005;36:1496–9.
[27] Hong S, Oh J-H, Park P-Y, Kim T-S, Kim S. Enhancement of motion image quality in LCD. Digest SID 2004;35:1353–5.
[28] Sluyterman A, Boonekamp E. Architecture choices in a scanning backlight for large LCD TVs. Digest SID 2005;36:996–9.
[29] Erwin JGJ, Bellers B, Penners M. Motion compensated frame rate conversion for motion blur reduction. Digest SID 2007;38:1454–7.
[30] Hong S, Berkeley B, Kim SS. Motion image enhancement of LCDs. In: ICIP (2), 2005. p. 17–20, doi: 〈http://dx.doi.org/10.1109/ICIP.2005.1529980〉.
[31] Kurita T. Motion-adaptive edge compensation to decrease motion blur of hold-type display. Digest SID 2005;36:1586–9.
[32] Klompenhouwer MA, Velthoven LJ. Motion blur reduction for liquid crystal displays: motion compensated inverse filtering. In: Proceedings of SPIE-IS&T electronic imaging, vol. 5308, 2004. p. 690–9.
[33] Jun Xia YS, Yin H. Motion adaptive deblurring filter for LCD. Displays 2009;30:27–31.
[34] Har-Noy S, Nguyen T. LCD motion blur reduction: a signal processing approach. IEEE Trans Image Process 2008;17(2):117–25.
[35] Har-Noy S, Nguyen TQ. LCD motion blur reduction using FIR filter banks. In: Proceedings of ICIP, 2009.
[36] Shannon CE. Communication in the presence of noise. In: Proceedings of the IRE, vol. 37, 1949. p. 10–21.
[37] Unser M. Sampling—50 years after Shannon. Proc IEEE 2000;88(4):569–87.
[38] Mitchell DP, Netravali AN. Reconstruction filters in computer-graphics. SIGGRAPH Comput Graphics 1988;22(4):221–8.
[39] Freeman WT, Adelson EH, Heeger DJ. Motion without movement. In: SIGGRAPH '91: proceedings of the 18th annual conference on computer graphics and interactive techniques, 1991. p. 27–30, doi: 〈http://doi.acm.org/10.1145/122718.122721〉.