A Shaped Temporal Filter Camera

Martin Fuchs, Tongbo Chen, Oliver Wang, Ramesh Raskar, Hans-Peter Seidel, and Hendrik P. A. Lensch

MPI–I–2009–4–003 June 2009

Authors' Addresses

Martin Fuchs
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Tongbo Chen
USC ICT
Marina del Rey, CA, USA

Oliver Wang
UC Santa Cruz
Santa Cruz, CA, USA

Ramesh Raskar
MIT Media Lab
Cambridge, MA, USA

Hans-Peter Seidel
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Hendrik P. A. Lensch
Ulm University
Ulm, Germany

Acknowledgements

This work 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).

Abstract

In the time domain, movie cameras typically perform a discrete sampling of real-world imagery. While the effect of discrete sampling in the spatial domain is well understood, the effect on temporal sampling during acquisition has been rarely addressed in the past, partially due to the lack of configurable hardware. In this article, we investigate the effect of different temporal sampling kernels as prefilters and show that extended, overlapping kernels can mitigate aliasing artifacts. Further, NPR effects, such as enhanced motion blur, can be achieved. In addition, online Fourier transforms in the temporal domain provide a novel tool for analyzing and visualizing time dependent effects. An integrated system, consisting of a 500 Hz high speed camera and a desktop computer with an off-the-shelf GPU, performs the necessary filtering or analysis in real-time and outputs the filtered result at video rates. We demonstrate the effect of different sampling kernels in creating enhanced movies and stills of fast motion.

Keywords

computational videography, computational photography, real-time, sampling, temporal filtering

Contents

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 (≈ 60Hz).

Since the dawn of cinematography, video and film cameras have performed a rather similar 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 prefiltering is applied prior to sampling the animation.

In this paper, we will discuss the construction of an optimal sampling filter given the characteristics of the output device. Furthermore, we will present a novel computational imaging system which allows for temporal pre-filtering to dampen temporal aliasing in real-time. 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, e.g. optimally pre-filtering for a given output kernel or artistically emphasizing or modulating motion blur. Furthermore, we can apply specialized filter banks for analyzing the signal in the Fourier domain, in order to understand and enhance video content based on its temporal behavior, e.g. emphasizing or deemphasizing motion.

Our system consists of a high speed camera coupled to a high performance GPU. 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

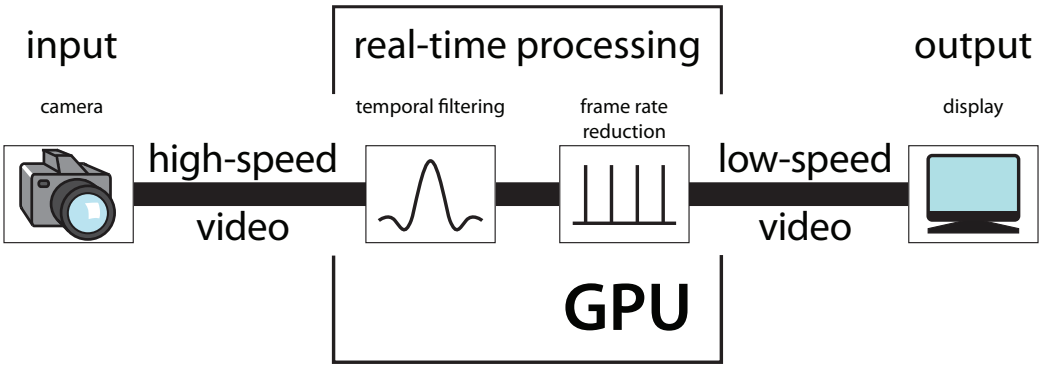


Figure 1.1: Processing pipeline of the proposed camera system.

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 (Section 3),
- a prototype system that performs real-time temporal filtering with overlapping kernels of arbitrary shapes (Section 4), and
- a Fourier camera that can perform online image processing in the temporal Fourier domain (Section 5) such as frequency analysis and motion enhancement.

We demonstrate the following real-time applications: suppression of temporal aliasing in video sequences, 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 [7, 17], to speed up the computation of animation sequences by frame-less rendering [3], and to faithfully create motion blur with photon mapping [5], to name a few. They can be synthesized using the accumulation buffer [11].

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 [4] 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 [1, 21], and to estimate and extend the motion in a scene [13]. 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 computer 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, e.g. [6]. In Section 4.3, 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 [10, 26]. Shechtman et al. [20] combined a set of video cameras to produce space-time super resolution videos which allowed for off-line temporal filtering. Bennett and McMillan [2] perform filtering on standard frame-rate

video to create time-lapse output. The propose a virtual shutter for additional effects. We will present an online system that requires only a single high-speed camera.

Wilburn et al. [25] 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. [18] 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 [12]. 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 [23, 15, 16]. 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.[8, 24]) offer real-time video manipulation but typically do not provide additional means to control the temporal filtering. Recently, the first consumer cameras appeared (e.g. 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.

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 some 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:

$$I(x, y) = \int_{-\infty}^{\infty} \Phi(x, y, t) \cdot m(t, x, y)\, dt. \tag{3.1}$$

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.

3.2 Imaging and Sampling Theory

Equation 3.1 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 then a device that generates a set of samples c_τ at discrete points in time, so that

$$c_\tau = \int_{-\infty}^{\infty} s(t) \cdot m(t - \tau)\, dt \tag{3.2}$$

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

$$\tilde{s}(t) = \sum_\tau c_\tau \cdot r(t - \tau) \tag{3.3}$$

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

We know from the work of Shannon [19], 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 $(c_t)_{t\in\mathbb{Z}}$ from which a perfect reconstruction is possible. Shannon's observations tell us that a perfect reconstruction

$$s(t) = \sum_\tau c_\tau \cdot r(t - \tau) = \sum_\tau \int_{-\infty}^{\infty} s(t) \cdot m(t - \tau)\, dt \cdot r(t - \tau) \tag{3.4}$$

is possible for the choice of

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

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 [22], 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 Equation 3.3). In the least squares sense, this is a projection into the function space spanned by the $(r_\tau)_{\tau\in\mathbb{Z}}$,

$$r_\tau(t) := r(t-\tau). \tag{3.5}$$

Unser [22] describes techniques to compute the kernel $m(t)$ that approximate this equation. 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 $\frac{\nu}{2}$, 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.

3.3 Kernel Shapes for Optimal Pre-Filtering

We will now discuss typical reconstruction functions $r(t)$ of the output device 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 Figure 3.1). We discretize each sampling interval $T = \frac{1}{\nu} = 1$ 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 behaviors that can be observed in real displays:

Box function (zero-order-hold) with width T and height 1, i.e. $r(t) = \text{rect}(t)$. 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. $r(t) = \text{tri}(t) = \text{rect}(t) * \text{rect}(t)$. 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 $\text{rect}(t)$ i times with itself, we can 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.

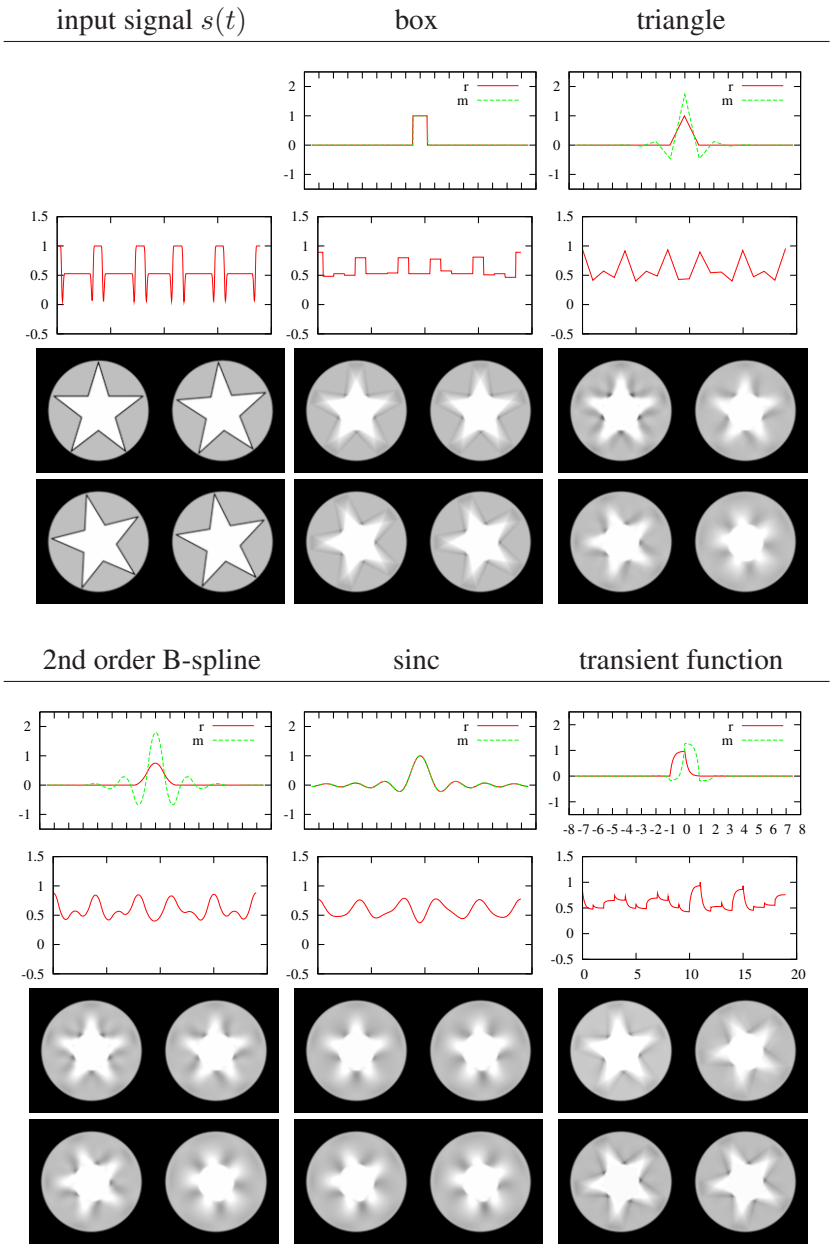


Figure 3.1: Filtering results for a spinning star. From top to down: shape of the pre-filter (green) and the reconstruction kernel (red), intensity profile for a single pixel, and reconstructed images for individual time steps ($t \in 0, 1/4, 1/2, 3/4$ clockwise starting top left) within one frame.

Sinc: $r(t) = \text{sinc}(t)$ 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 $\text{rect}(t)$ 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, e.g. as dampened by a capacitor. The filter

$$r(t) = \begin{cases} e^{-\lambda t} & \text{if } 0 \leq t < 1 \\ 1 - e^{-\lambda(t+1)} & \text{if } -1 \leq t < 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

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 $T \in \{0, 1\}$.

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 comes close to a rectangular kernel $m(t) = \text{rect}(t/w)$ with width w. In the case that the kernel spans the entire duration between two frames ($w = 1$), and assuming an unrealistic output device with $r(t) = \text{rect}(t)$, this sampling kernel would be L_2-optimal. Most often, the exposure time is however much shorter than the frame duration to avoid saturation, and the system produces severe aliasing effects such as "wagon-wheel moving backwards" illusion or jagged, discontinuous motion (see the 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.

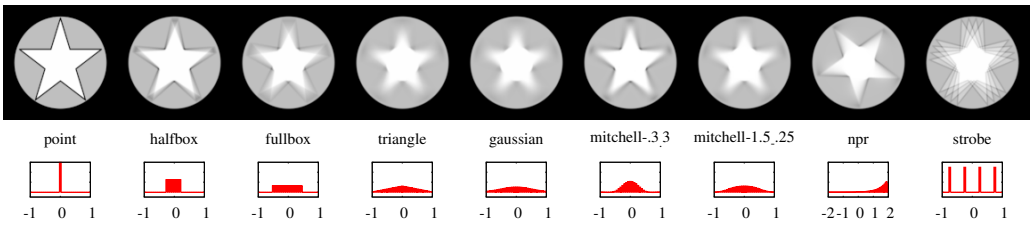


Figure 3.2: Excerpt from a simulated sequence of a spinning star and the generating kernels (bottom).

3.4 Evaluation of Further Kernel Types

In the following paragraphs, we will discuss a selection of other ad-hoc kernel shapes and their effects. Unlike in the previous discussion we will not take the reconstruction into account and only perform a shaped exposure filtering for visually pleasing results. Figure 3.2 demonstrates 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 $w = \frac{T}{2}$ 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 are perceptually close in appearance, providing a smooth, continuous transition between subsequent frames. Each individual output frame however looks probably too smooth.

In their work on reconstruction filters [14], **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 non-linear exponential function $m'(t) = e^{(-1+t)\cdot 2}$ and blurred 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 4 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 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.

4 Real-Time Processing System

We will now introduce the design aspects of our hardware and software prototype.

4.1 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 480Hz at a resolution of 1000×1024 or 500Hz 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 60Hz. As an alternative capturing device, we employ a Casio Exilim Pro EX-F1 camera to record high speed video at 300Hz.

4.2 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 towards the GPU and perform all operations on these single-channel images. Only for display, we run demosaicing by bi-linear interpolation and subtract the black frame at 60Hz.

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

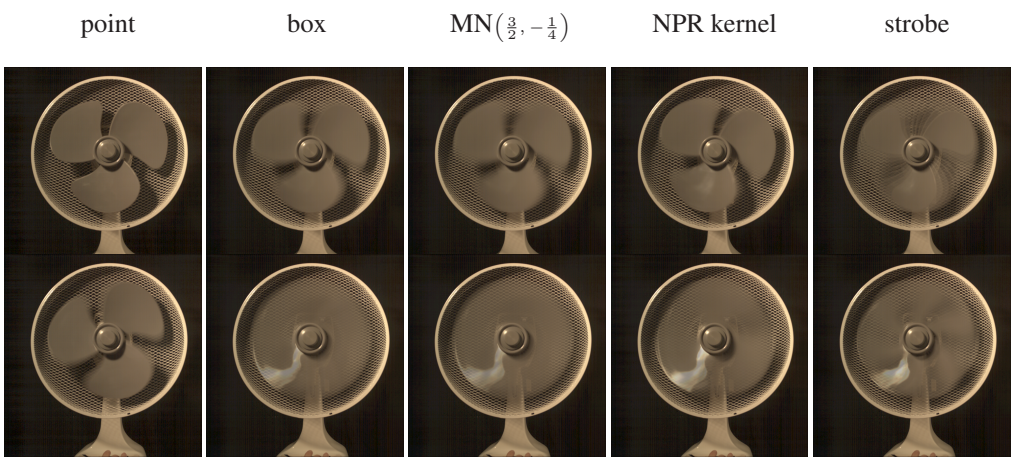


Figure 4.1: Various filters applied to the video of a spinning fan at two different velocities. Aliasing effects are visible in the accompanying video for all filters except for the MN filter.

4.3 Results for Rigid Body Motion

Figure 4.1 shows the effect of different filtering kernels on the repeating motion of a rotating fan. Using a single 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.

4.4 Results for Stochastic Processes

In Figure 4.2 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 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.

point full-frame box Mitchell-Netrav. $(\frac{3}{2}, -\frac{1}{4})$ NPR kernel strobe

Figure 4.2: Filtering results for stochastic processes. Depending on the applied filter individual particles or the motion direction is visible. The non-linearly increasing filter combines the details of the point filter while indicating the motion direction.

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.

5 Fourier Camera

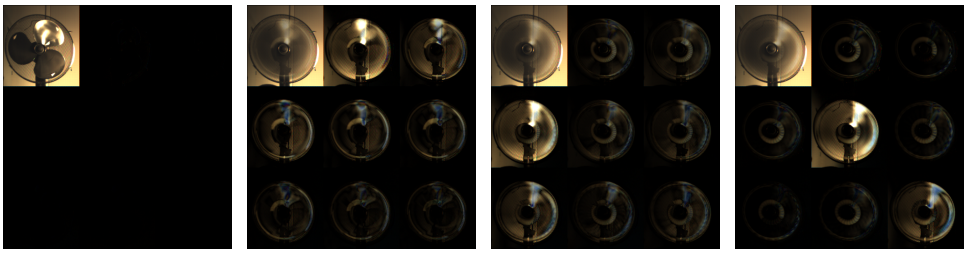
The ability to perform the convolution of high speed video material in real-time makes it possible to visualize periodic or non-periodic movement in a novel way: by performing a discrete per-pixel temporal Fourier transform on the input. Per-pixel oscillations can be observed in different frequency-bands, 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 500Hz. We compute 9 bands with a resolution of 240×256 each at 50Hz. A different Fourier transform is calculated for each frame.

5.1 Fourier Domain Applications

Figure 5.1 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 (Figure 5.2). 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 Figure 5.3. 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; the spatial extent and visual contrast is influenced by the selected frequencies.

Figure 5.4 shows the frequency boosting effect on thin plant leaves moving



stationary spinning up slow speed high speed

Figure 5.1: 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.

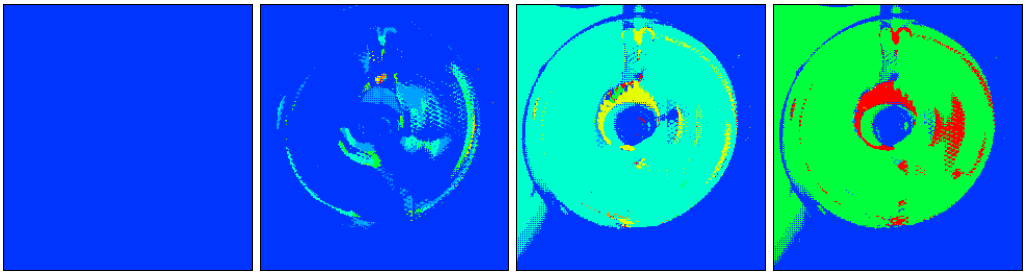


Figure 5.2: Color-coded maximum frequency of the fan of Figure 5.1 spinning at four different velocities.

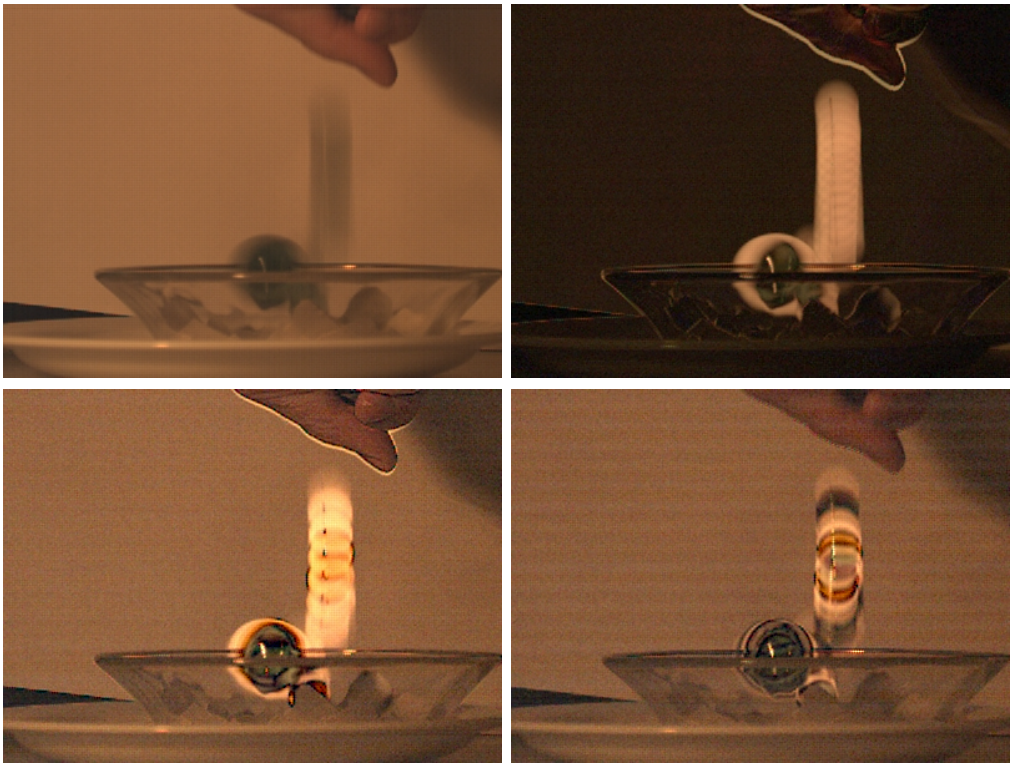


Figure 5.3: Modified appearance of a video sequence in Fourier space: standard Gaussian filter (top left), absolute values after removing the DC band (top right), boosting low frequencies (bottom left), and boosting high frequencies (bottom right).

Figure 5.4: In contrast to a plain Gaussian filter kernel (left), boosting frequency components emphasizes the motion.

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 [9] 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.

6 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 (< 2ms), 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 [1, 21] as well as our own footage successfully show that the after integration of the recorded frame this effect is significantly 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@500Hz compared to the maximum resolution of the camera (1280×1024) and on the GPU when integrating into multiple output frames.

7 Conclusion

In this paper we have 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 will become available in consumer cameras in the near future.

We argue that in order to prevent severe temporal aliasing it is paramount 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.

In the accompanying video we demonstrated the effectiveness of the proposed temporal filters on producing artifact reduced videos as well as more expressive stills for stochastic motion events such as water falls or flames. A user study would be necessary to objectively determine an optimal filter kernel.

Some of the filters could 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 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 [2], shown in Figure 7.1.

As our system allows to freely change and control the shape of the temporal

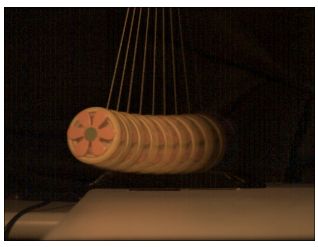


Figure 7.1: A single frame of a video q enhanced in real-time with motion trails.

filter during online recording, it can not 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.

Bibliography

[1] E. P. Bennett and L. McMillan. Video enhancement using per-pixel virtual exposures. *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, 24(3):845–852, 2005.

[2] E. P. Bennett and L. McMillan. Computational time-lapse video. *ACM Transactions on Graphics (Proc. SIGGRAPH 2007)*, 26(3):102, 2007.

[3] G. Bishop, H. Fuchs, L. McMillan, and E. J. S. Zagier. Frameless rendering: double buffering considered harmful. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 175–176, 1994.

[4] G. J. Brostow and I. Essa. Image-based motion blur for stop motion animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 561–566, 2001.

[5] M. Cammarano and H. W. Jensen. Time dependent photon mapping. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*, pages 135–144, 2002.

[6] D. Collins and R. R. Bruce, editors. *Seeing the Unseen: Dr. Harold E. Edgerton and the Wonders of Strobe Alley*. MIT Press, 1994.

[7] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. pages 77–85, 1998.

[8] FastVision. *FastCamera40*, 2008. http://www.fast-vision.com/cameras/camera40.html.

[9] W. T. Freeman, E. H. Adelson, and D. J. Heeger. Motion without movement. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 27–30, New York, NY, USA, 1991. ACM.

[10] H. Gao, J. Zhang, S. Yoshikado, and T. Aruga. Photorefractive low-pass temporal filter. *Optics Communications*, 203(3–6):363–369, 2002.

[11] P. Haeberli and K. Akeley. The accumulation buffer: hardware support for high-quality rendering. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 309–318, New York, NY, USA, 1990. ACM.

[12] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. *ACM Transactions on Graphics (Proc. SIGGRAPH 2008)*, 27(3):711–719, 2008.

[13] C. Liu, A. Torralba, W. T. Freeman, F. Durand, and E. H. Adelson. Motion magnification. In *ACM Transactions on Graphics (Proc. SIGGRAPH 2005)*, pages 519–526, 2005.

[14] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer-graphics. *SIGGRAPH Comput. Graph.*, 22(4):221–228, 1988.

[15] Motion Analysis Corporation. *Hawk-I Digital System*, 2006.

[16] Optotrak. *NDI Optotrak Certus Spatial Measurement*, 2007. http://www.ndigital.com/certus.php.

[17] M. Potmesil and I. Chakravarty. Modeling motion blur in computer-generated images. *SIGGRAPH Comput. Graph.*, 17(3):389–399, 1983.

[18] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics (Proc. SIGGRAPH 2006)*, 25(3):795–804, 2006.

[19] C. E. Shannon. Communication in the presence of noise. In *Proc. IRE*, volume 37, pages 10–21, 1949.

[20] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 753–768, 2002.

[21] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis. Synthetic shutter speed imaging. *Computer Graphics Forum*, 26(3):591–598, 2007.

[22] M. Unser. Sampling—50 Years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.

[23] Vicon Peak. *Camera MX 40*, 2006. http://www.vicon.com/products/mx40.html.

[24] Vision Components. *VC4472*, 2008.

[25] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24(3):765–776, Aug. 2005.

[26] D. Yang, S. Kumar, and H. Wang. Temporal filtering using time lenses for optical transmission systems. *Optics Communications*, 281(2):238–247, 2008.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available via WWW using the URL `http://www.mpi-inf.mpg.de`. If you have any questions concerning WWW access, please contact `reports@mpi-inf.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-inf.mpg.de`

MPI-I-2009-RG1-002	P. Wischnewski, C. Weidenbach	Contextual rewriting
MPI-I-2009-5-006	S. Bedathur, K. Berberich, J. Dittrich, N. Mamoulis, G. Weikum	Scalable phrase mining for ad-hoc text analytics
MPI-I-2009-5-004	N. Preda, F.M. Suchanek, G. Kasneci, T. Neumann, G. Weikum	Coupling knowledge bases and web services for active knowledge
MPI-I-2009-5-003	T. Neumann, G. Weikum	The RDF-3X engine for scalable management of RDF data
MPI-I-2008-RG1-001	A. Fietzke, C. Weidenbach	Labelled splitting
MPI-I-2008-5-004	F. Suchanek, M. Sozio, G. Weikum	SOFI: a self-organizing framework for information extraction
MPI-I-2008-5-003	F.M. Suchanek, G. de Melo, A. Pease	Integrating Yago into the suggested upper merged ontology
MPI-I-2008-5-002	T. Neumann, G. Moerkotte	Single phase construction of optimal DAG-structured QEPs
MPI-I-2008-5-001	F. Suchanek, G. Kasneci, M. Ramanath, M. Sozio, G. Weikum	STAR: Steiner tree approximation in relationship-graphs
MPI-I-2008-4-003	T. Schultz, H. Theisel, H. Seidel	Crease surfaces: from theory to extraction and application to diffusion tensor MRI
MPI-I-2008-4-002	W. Saleem, D. Wang, A. Belyaev, H. Seidel	Estimating complexity of 3D shapes using view similarity
MPI-I-2008-1-001	D. Ajwani, I. Malinger, U. Meyer, S. Toledo	Characterizing the performance of Flash memory storage devices and its impact on algorithm design
MPI-I-2007-RG1-002	T. Hillenbrand, C. Weidenbach	Superposition for finite domains
MPI-I-2007-5-003	F.M. Suchanek, G. Kasneci, G. Weikum	Yago : a large ontology from Wikipedia and WordNet
MPI-I-2007-5-002	K. Berberich, S. Bedathur, T. Neumann, G. Weikum	A time machine for text search
MPI-I-2007-5-001	G. Kasneci, F.M. Suchanek, G. Ifrim, M. Ramanath, G. Weikum	NAGA: searching and ranking knowledge
MPI-I-2007-4-008	J. Gall, T. Brox, B. Rosenhahn, H. Seidel	Global stochastic optimization for robust and accurate human motion capture
MPI-I-2007-4-007	R. Herzog, V. Havran, K. Myszkowski, H. Seidel	Global illumination using photon ray splatting
MPI-I-2007-4-006	C. Dyken, G. Ziegler, C. Theobalt, H. Seidel	GPU marching cubes on shader model 3.0 and 4.0
MPI-I-2007-4-005	T. Schultz, J. Weickert, H. Seidel	A higher-order structure tensor
MPI-I-2007-4-004	C. Stoll, E. de Aguiar, C. Theobalt, H. Seidel	A volumetric approach to interactive shape editing
MPI-I-2007-4-003	R. Bargmann, V. Blanz, H. Seidel	A nonlinear viseme model for triphone-based speech synthesis
MPI-I-2007-4-002	T. Langer, H. Seidel	Construction of smooth maps with mean value coordinates
MPI-I-2007-4-001	J. Gall, B. Rosenhahn, H. Seidel	Clustered stochastic optimization for object recognition and pose estimation
MPI-I-2007-2-001	A. Podelski, S. Wagner	A method and a tool for automatic veriication of region stability for hybrid systems
MPI-I-2007-1-003	A. Gidenstam, M. Papatriantafilou	LFthreads: a lock-free thread library
MPI-I-2007-1-002	E. Althaus, S. Canzar	A Lagrangian relaxation approach for the multiple sequence alignment problem
MPI-I-2007-1-001	E. Berberich, L. Kettner	Linear-time reordering in a sweep-line algorithm for algebraic curves intersecting in a common point

MPI-I-2006-5-006	G. Kasnec, F.M. Suchanek, G. Weikum	Yago - a core of semantic knowledge
MPI-I-2006-5-005	R. Angelova, S. Siersdorfer	A neighborhood-based approach for clustering of linked document collections
MPI-I-2006-5-004	F. Suchanek, G. Ifrim, G. Weikum	Combining linguistic and statistical analysis to extract relations from web documents
MPI-I-2006-5-003	V. Scholz, M. Magnor	Garment texture editing in monocular video sequences based on color-coded printing patterns
MPI-I-2006-5-002	H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum	IO-Top-k: index-access optimized top-k query processing
MPI-I-2006-5-001	M. Bender, S. Michel, G. Weikum, P. Triantafilou	Overlap-aware global df estimation in distributed information retrieval systems
MPI-I-2006-4-010	A. Belyaev, T. Langer, H. Seidel	Mean value coordinates for arbitrary spherical polygons and polyhedra in $\mathbb{R}^3$
MPI-I-2006-4-009	J. Gall, J. Potthoff, B. Rosenhahn, C. Schnoerr, H. Seidel	Interacting and annealing particle filters: mathematics and a recipe for applications
MPI-I-2006-4-008	I. Albrecht, M. Kipp, M. Neff, H. Seidel	Gesture modeling and animation by imitation
MPI-I-2006-4-007	O. Schall, A. Belyaev, H. Seidel	Feature-preserving non-local denoising of static and time-varying range data
MPI-I-2006-4-006	C. Theobalt, N. Ahmed, H. Lensch, M. Magnor, H. Seidel	Enhanced dynamic reflectometry for relightable free-viewpoint video
MPI-I-2006-4-005	A. Belyaev, H. Seidel, S. Yoshizawa	Skeleton-driven laplacian mesh deformations
MPI-I-2006-4-004	V. Havran, R. Herzog, H. Seidel	On fast construction of spatial hierarchies for ray tracing
MPI-I-2006-4-003	E. de Aguiar, R. Zayer, C. Theobalt, M. Magnor, H. Seidel	A framework for natural animation of digitized models
MPI-I-2006-4-002	G. Ziegler, A. Tevs, C. Theobalt, H. Seidel	GPU point list generation through histogram pyramids
MPI-I-2006-4-001	A. Efremov, R. Mantiuk, K. Myszkowski, H. Seidel	Design and evaluation of backward compatible high dynamic range video compression
MPI-I-2006-2-001	T. Wies, V. Kuncak, K. Zee, A. Podelski, M. Rinard	On verifying complex properties using symbolic shape analysis
MPI-I-2006-1-007	H. Bast, I. Weber, C.W. Mortensen	Output-sensitive autocompletion search
MPI-I-2006-1-006	M. Kerber	Division-free computation of subresultants using bezout matrices
MPI-I-2006-1-005	A. Eigenwillig, L. Kettner, N. Wolpert	Snap rounding of Bézier curves
MPI-I-2006-1-004	S. Funke, S. Laue, R. Naujoks, L. Zvi	Power assignment problems in wireless communication
MPI-I-2005-5-002	S. Siersdorfer, G. Weikum	Automated retraining methods for document classification and their parameter tuning
MPI-I-2005-4-006	C. Fuchs, M. Goesele, T. Chen, H. Seidel	An emperical model for heterogeneous translucent objects
MPI-I-2005-4-005	G. Krawczyk, M. Goesele, H. Seidel	Photometric calibration of high dynamic range cameras
MPI-I-2005-4-004	C. Theobalt, N. Ahmed, E. De Aguiar, G. Ziegler, H. Lensch, M.A. Magnor, H. Seidel	Joint motion and reflectance capture for creating relightable 3D videos
MPI-I-2005-4-003	T. Langer, A.G. Belyaev, H. Seidel	Analysis and design of discrete normals and curvatures
MPI-I-2005-4-002	O. Schall, A. Belyaev, H. Seidel	Sparse meshing of uncertain and noisy surface scattered data