Real-time Apparent Resolution Enhancement for Head-mounted Displays

HAEBOM LEE, Saarland University, MMCI
PIOTR DIDYK, Saarland University, MMCI, MPI Informatik, and Università della Svizzera italiana

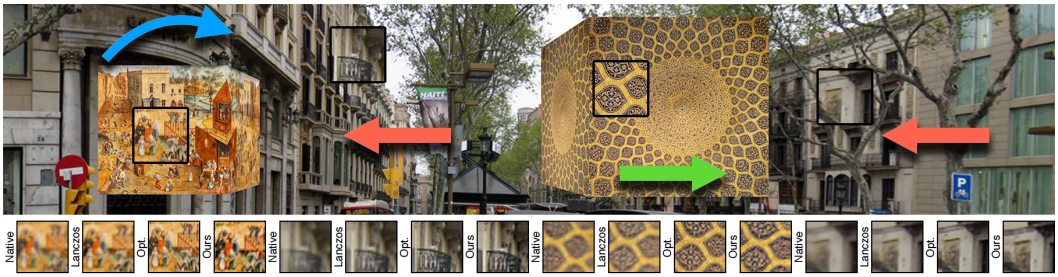

Fig. 1. Hardware limitations of novel head-mounted displays make it hard to reproduce highly detailed scenes. Our real-time apparent resolution enhancement method can enhance the perceived resolution of current display setups using a simple filtering step. The figure represents a 3D scene. The arrows indicate how different portions of the image move on the screen. The insets present a reproduction of fine spatial details for different techniques. Image courtesy: Thom Quine, Phillip Maiwald

The insufficient pixel density of current head-mounted displays is one of the major obstacles in achieving immersive and fully engaging experiences. It is possible to overcome this physical limitation for moving content using software techniques. To this end, previous techniques utilized high-framerate displays and optimized for low-resolution images that, when shown on a display, can significantly increase the apparent spatial resolution. However, so far, all the proposed techniques require expensive optimization, which makes the techniques unsuitable for real-time applications. To overcome this problem, we present a novel method that can improve apparent resolution of such displays in real time. We replace expensive optimizations with a two-step filtering approach. Due to the efficiency of our technique, we can account not only for the motion in the scene but also for any motion in the perceived image introduced by movement of a user. This greatly extends the range of situations where the resolution enhancement can be achieved. In this paper, we present the derivation of the motion-flow-dependent filters and how they can be applied to increase the perceived resolution. To evaluate the performance of our technique, we conducted a user experiment which compares our method to alternative solutions regarding perceived resolution as well as overall quality and demonstrates the advantages of our technique.

CCS Concepts: • **Human-centered computing** → **Displays and imagers**; • **Computing methodologies** → **Rendering**; **Mixed / augmented reality**; **Perception**; **Virtual reality**;

Additional Key Words and Phrases: spatial resolution enhancement, display, perception, virtual reality (VR), augmented reality (AR), head-mounted display (HMD)

2577-6193/2018/5-ART19 $15.00
https://doi.org/10.1145/3203202

ACM Reference Format:
Haebom Lee and Piotr Didyk. 2018. Real-time Apparent Resolution Enhancement for Head-mounted Displays. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1, Article 19 (May 2018), 15 pages. https://doi.org/10.1145/3203202

1 INTRODUCTION

The goal of today's display devices is to reproduce high fidelity content. To this end, displays have to provide an excellent reproduction of brightness, color, and fine details. For most recent devices which are capable of presenting stereoscopic images, a reproduction of depth cues also becomes essential. More specifically, in addition to the spatial resolution, displays have to provide high angular resolution to correctly address binocular disparity and accommodation. In the context of new virtual and augmented reality glasses (VR/AR), the resolution becomes a significant issue. While the development of these technologies has benefited in recent years from inexpensive high-resolution display devices, such as smartphones, these were not primarily designed for near-eye scenarios. As a result, the maximum spatial frequencies that can be reproduced by current head-mounted displays is significantly below what the human visual system (HVS) can perceive. More importantly, while it is possible to render or acquire high-resolution content, it cannot be faithfully reproduced on novel display devices such as VR and AR headsets. This hampers the adoption of these devices as the visual quality is often insufficient for users.

The problem of low spatial resolution in current near-eye displays is widely acknowledged [Road to VR 2017]. Existing commercial headsets, such as the HTC Vive or Oculus Rift, provide an average pixel density of approximately $11\,^{\mathrm{px}}/_{\mathrm{deg}}$ [Road to VR 2017]. From the sampling point of view, this allows for reconstructing spatial frequencies of $5.5\,^{\mathrm{cycles}}/_{\mathrm{deg}}$. In practice, due to the lens distortions, the resolution is not equal across the screen, and it reaches a maximum in the center of the screen. However, even the slightly higher resolution in the center of the screen is significantly below what human eyes can perceive. For comparison, the highest anatomically determined spatial frequency that can be resolved by a human observer according to the density of cones in the fovea [Curcio et al. 1990] and Nyquist's theorem is roughly $60\,^{\mathrm{cycles}}/_{\mathrm{deg}}$.

While the most intuitive solution for the resolution mismatch is improving the resolution of display panels, it has recently been demonstrated that the apparent resolution can also be significantly improved using high-framerate displays and software techniques [Berthouzoz and Fattal 2012a; Didyk et al. 2010; Stengel et al. 2013; Templin et al. 2011]. The key idea of these methods is to exploit temporal integration of the HVS and the fact that the HVS closely follows moving objects. These techniques exploit the motion in the scene and optimize for low-resolution sub-frames which after the integration give an impression of looking at high-resolution images. Although the methods have been proven to be effective, the resolution enhancement comes at a significant computational cost spent on optimizing frames of animation, which becomes prohibitively expensive in the context of real-time applications.

To address this problem, we propose a real-time apparent resolution enhancement. Our method is based on the previous solutions but avoids solving expensive optimization, which is replaced by a two-step filtering process. As a result, the technique provides a resolution boost at sufficiently high-framerates, which enable an application to real-time rendering. The core of our technique lies in computing a velocity dependent filter bank based on a set of previously optimized image sequences. The filters, when applied to high-resolution content, mimic previously used optimizations. Since our technique can be performed in real time as part of a rendering pipeline, it can account not only for existing motion in the scene, but also any motion in the image space introduced by a user. This enables resolution enhancement during head movements and walk-through scenarios, which are very common in VR applications, but impossible to address using previous solutions. In this paper, we demonstrate the derivation of the filters and how to apply them to rendered

content. Additionally, we conduct perceptual experiments which demonstrate the performance of our technique both on standard desktop screens and a VR headset. In this paper, we make the following contributions:

- a simplified formulation of apparent resolution optimization,
- derivation of a velocity-dependent filter bank that is used to improve apparent resolution,
- application of filters to real-time rendering while accounting for motion in the scene and head movements, and
- an evaluation of the technique in a user experiment.

2 RELATED WORK

Several attempts were made to enhance display resolution using advanced hardware designs or software techniques [Masia et al. 2013]. In this section, we briefly introduce relevant previous work.

2.1 Hardware designs

A straightforward approach to increase display resolution is to increase pixel density. This can be difficult and expensive; therefore, several works demonstrated that it is possible to achieve significant resolution gains by exploiting existing lower-resolution hardware. To increase the perceived resolution, such techniques usually rely on a spatial or temporal superposition of low-resolution images.

Damera-Venkata and Chang [2009] presented an approach using spatial superposition. They combined multiple standard low-resolution projectors to enhance the resolution of displayed output. The key idea was to introduce a sub-pixel offset to each projected image. The small misalignment led to increased resolution of the combined result. The paper provides a theoretical analysis of derivation of the low-resolution images and proposes an optimization procedure for deriving projected images for arbitrary offsets. Furthermore, the authors suggested a filtering method for real-time applications which generates the images comparable to the optimization results. More recently, a similar idea of spatial multiplexing of low-resolution images was introduced in the context of head-mounted displays. Heide et al. [2014] demonstrated that spatial resolution can be increased by cascading a pair of LCD panels with a sub-pixel offset. Similarly to the previous solution, they use an optimization procedure to decompose a high-resolution input image into two images that are shown on the superimposed panels.

Instead of superimposing multiple low-resolution images spatially, it is possible to use temporal multiplexing and rely on temporal averaging performed by the HVS. One of the first such solutions was proposed by Allen and Ullichney [2005]. They demonstrated an optical system which projects every two consecutive images with a sub-pixel offset. In this case, similarly to other techniques, such a spatial misalignment improves the resolution of the final image. Berthouzoz and Fattal [2012b] further developed this idea and presented a method that can increase the resolution by vibrating a high-framerate display. Their design involved display vibration carefully synchronized with the refresh cycle of the screen panel. Given the fast rotational movement of the display, they solved an optimization problem to derive images that are displayed on the panel. More recently, similar ideas were discussed by Kading and Straub [2015] in the context of head-mounted setups. They proposed two display designs. One consisted of two superimposed display units with a half-pixel offset and the other introduced circular motion with a one-pixel diameter to the display panel. Similarly to the method of Berthouzoz and Fattal [2012b], the design exploits a high-framerate display and the temporal integration of the HVS. Despite possible resolution enhancement, the authors mention that the solution is not practical due to the need for introducing motion to complex electronics.

2.2 Apparent resolution enhancement

While the above hardware solutions require hardware modifications which are not always easy to realize, it has been demonstrated that a careful optimization of images shown on a screen can lead to significant increase in apparent resolution.

The first group of such techniques exploits the subpixel layout of a display panel. They take advantage of the lower sensitivity of the HVS to chromatic information than to luminance. This allows introducing high-frequency color patterns which increase the apparent resolution without introducing color artifacts. An early example is the work of Platt [2000] who derived an optimal filtering strategy for LCD panels used for ClearType fonts. Later, Messig and Kerofsky [2006] proposed an optimization procedure that could handle various sub-pixel layouts. More recently, the sub-pixel filtering was incorporated into GPU-based multisample antialiasing and resulted in a subpixel rendering at a very low cost [Engelhardt et al. 2014].

A different approach to resolution enhancement was proposed by Didyk et al. [2010]. They utilized properties of the HVS to generate low-resolution sub-frames that, when shown on a high-framerate display, can be perceived as a high-resolution image. The technique exploits the fact that human eyes consistently follow a moving image on a display using smooth-pursuit eye motion (SPEM), and that several consecutive frames are integrated due to the temporal integration performed by the HVS. These observations allow them to propose a simple model that predicts the perceived image given a sequence of sub-frames shown on a screen. Later, they formulated an optimization procedure that decomposes a high-resolution input to lower-resolution sub-frames that create an impression of looking at the high-resolution sequence when displayed on a high-framerate display and combined on the retina. While the original technique was demonstrated for linearly moving images, Templin et al. [2011] extended it to animations by taking the motion already existing in a video sequence into account. The technique assumed that an observer locally follows the moving objects, and the motion is well approximated by the optical flow of the scene. It has also been demonstrated that apparent resolution can be combined with a super-resolution technique [Berthouzoz and Fattal 2012a]. This avoids the need for a high-resolution input since fine details can be extracted directly from a low-resolution image sequence. One of the major limitations of such apparent resolution techniques is that the resolution enhancement depends on the motion in the scene. The problem was addressed by Stengel et al. [2013] who proposed to modify the input content, i.e., introduce additional movement, to maximize the benefits of the resolution enhancement.

While the techniques exploiting temporal integration of the HVS can provide apparent resolution enhancement, the optimizations that they employ are prohibitively expensive for real-time applications. In this work, we overcome this limitation, by demonstrating that the costly optimizations can be replaced by a filtering step which adds little computational overhead to the rendering.

3 APPARENT RESOLUTION ENHANCEMENT

In this work, we aim at replacing expensive optimization that was used for apparent resolution enhancement with simple filtering steps. To this end, we will first provide an overview of previous techniques on which we base our solution, and reformulate them to enable the derivation of our filtering.

3.1 Optimization approach

Didyk et al. [2010] considered a simple problem in which an image is moving across a screen with a constant velocity. This work assumes that the HVS acts as a temporal box filter which averages light intensity over a short period of time. Consequently, the authors model the response of a single

receptor on a retina as:

$$r = \int_0^T I(p(t), t)dt, \tag{1}$$

where p denotes the position on the screen from which the receptor r receives the light as a function of time t, I describes the screen intensity at a given point and time, and T is a sufficiently short time interval within which the temporal integration happens [Kalloniatis and Luu 2007]. Since a screen displays a single image for an extended period of time, Eq. 1 can be expressed in a discrete form:

$$r = \sum_{i,j,k} w_{i,j}^k \cdot I_{i,j}^k, \tag{2}$$

where $I_{i,j}^k$ describes the intensity of the k-th image at position (i, j) and $w_{i,j}^k$ are corresponding weights which encode how long a given receptor r was observing pixel $I_{i,j}^k$. Templin et al. [2011] provided a more formal definition of the weights using indicator functions:

$$w_{i,j}^k = \frac{1}{|p|} \int_0^T \mathbf{1}_{i,j}(p(t))\, \mathbf{1}_k(t)dt. \tag{3}$$

The indicator function $\mathbf{1}_{i,j}$ becomes 1 if the position $p(t)$ is within the pixel at (i, j), while $\mathbf{1}_k$ outputs 1 if the k-th image is displayed at time t. The integrated value is then normalized by $|p|$, the total length of the path.

Using the above formulation for modeling the response of one photoreceptor, it is possible to predict the whole image perceived by an observer. To this end, Didyk et al. [2010] assumed that the retina is composed of a large number of receptors located on a grid. This way, the responses of all receptors can be associated with a high-resolution retinal image. With this simplification, the temporal integration described above can be formulated as a system of linear equations predicting a retinal image I_R:

$$I_R = W \cdot \mathbf{x}, \tag{4}$$

where W encodes weights $w_{i,j}^k$ from Eq. 2, and $\mathbf{x}$ consists of all images $\{I_k\}$ shown on a screen in a vectorized form. To enhance the apparent resolution, Didyk et al. [2010] formulated an optimization problem based on Eq. 4. The optimization decomposes a high-resolution image I_H into a set of sub-frames $\{I_k\}$ that, when shown on a screen, lead to a high-resolution experience. More formally:

$$\begin{aligned} \widetilde{\mathbf{x}} &= \underset{\mathbf{x}}{\operatorname{argmin}} \|W\mathbf{x} - I_H\|^2 \\ &\text{subject to : } \mathbf{0} \le \mathbf{x} \le \mathbf{1} \end{aligned}, \tag{5}$$

where I_H is the goal high-resolution image, W encodes the integration weights given by a previously assumed motion of the image, and $\widetilde{x}$ represents the set of resulting sub-frames. Following the findings regarding the temporal integration of the HVS [Kalloniatis and Luu 2007], the authors fixed T to 1/40 s, which allowed them to optimize for three sub-frames shown on 120 Hz display. Templin et al. [2011] further developed this idea to exploit existing motion in the scene. The work involved solving a similar optimization, but locally.

3.2 Filtering Formulation

Since the HVS is assumed to integrate the signal over a short period of time, the prediction of the retinal image (Eq. 4) can be expressed as a convolution of sub-frames. More specifically, during a period of one frame, the image stays constant on the display, but the eyes track moving content. As a result, the eyes blur the image with a velocity-dependent 1D filter. Additionally, since we consider high-framerate displays, the HVS will average results of the convolution from several sub-frames.

It is important to note that, since the eyes follow moving objects, the filtered sub-frames have to be shifted to simulate their alignment on the retina due to SPEM. The process can be written as:

$$I_R = \sum_k G_v * T(I_k, -kv), \tag{6}$$

where v is the velocity of the image expressed in pixels per frame, G_v is a velocity-specific box filter which simulates the blur introduced by moving eyes during the period of one frame, and $T(I_k, -kv)$ denotes the translation of each sub-frame to simulate the SPEM and to align the sub-frames with respect to each other. Since we consider short time intervals, a constant velocity can be assumed. Consequently, G_v is also constant, and the above equation can be rewritten as:

$$I_R = G_v * \sum_k T(I_k, -kv) \tag{7}$$

Similarly to the original optimization approach (Eq. 5), we can use Eq. 7 to define a new optimization which decomposes a high-resolution image I_H into a set of sub-frames $\{I_k\}$ based on the image velocity v:

$$\{I_k\} = \underset{\{\mathbf{I_k}\}}{\operatorname{argmin}} \left| \left(G_v * \sum\nolimits_k I'_k \right) - I_H \right|, \tag{8}$$

where $I'_k = T(I_k, -kv)$. Instead of solving this problem, we propose to approximate the solution using two filtering steps. First, we invert filter G_v and apply it to the input high-resolution image, obtaining $\widetilde{I_H} = G_v^{-1} * I_H$, which is similar to motion-compensated inverse filtering (MCIF) [Klompenhouwer and Velthoven 2004]. After that, our problem is simplified to:

$$\{I_k\} = \underset{\{\mathbf{I_k}\}}{\operatorname{argmin}} \left| \sum\nolimits_k I'_k - \widetilde{I_H} \right|. \tag{9}$$

While solving this optimization with constraints that the values of $\{I_k\}$ lie in the range of $(0, 1)$ would provide an optimal solution, such an approach would also prohibit performance at high-framerates. However, it has been demonstrated that solution to such a problem can be approximated using a set of filters [Damera-Venkata and Chang 2009] (Sec. 2). Consequently, instead of solving the problem defined in Eq. 9, we find a set of filters $\{D_{v,k}\}$ (Section 4.2) which when applied to $\widetilde{I_H}$ provide an approximate solution and define the solution to the problem in Eq. 8 as:

$$I_k = \downarrow T(D_{v,k} * G_v^{-1} * I_H, kv), \tag{10}$$

where $\downarrow$ denotes an operation of subsampling an image to the display resolution.

4 RESOLUTION ENHANCEMENT FILTERING

In this section, we show how to compute filters G_v^{-1} and $D_{v,k}$ and how they can be applied to improve the apparent resolution.

4.1 Derivation of G_v^{-1} Filters

Filter G_v is a 1D box filter applied to an image along a motion direction, whose spatial support is equal to the magnitude of velocity v. Since a box filter has a frequency response which is a *sinc* function (Fig. 2, blue), its direct inverse (Fig. 2, green) contains very large values which prevent simple inversion. To solve this problem, we follow the solution proposed by Klompenhouwer and Velthoven [2004] who perform a similar inversion to reduce blur caused by LCDs. Instead of directly

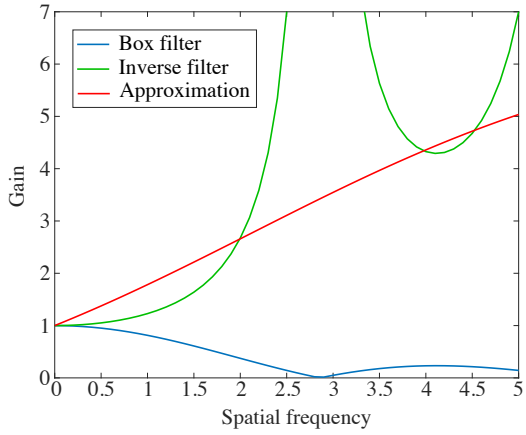

Fig. 2. Frequency responses of a box filter G_v (blue), its direct inverse (green), and our approximated inverse G_v^{-1} (red).

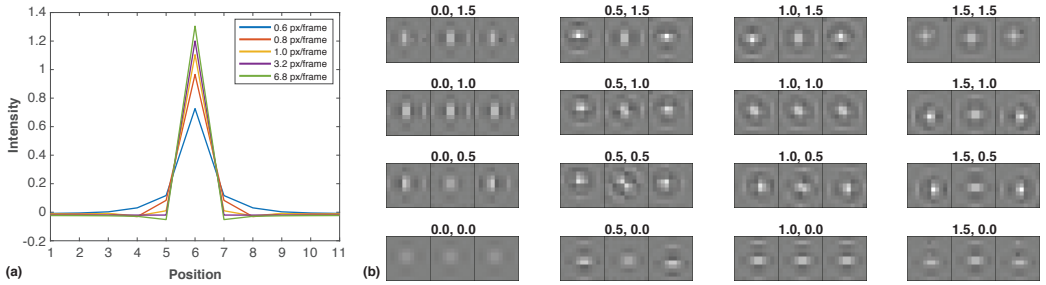

Fig. 3. Examples of (a) G_v^{-1} filters for different motion velocities and (b) sub-frame generation kernels $D_{v,k}$ for $k = 3$.

inverting the filter in the frequency domain, we approximate the inverse of the *sinc* function with a sigmoid function (Fig. 2, red) and use this as the desired response of G_v^{-1}. We use a sigmoid function:

$$a\left(0.5 - \frac{1}{1 + e^{bx-c}}\right) + d,$$

and optimize for its parameters (i.e., a, b, c, d) for each velocity separately. To speed up an application of these filters, we precompute the kernels for velocities ranging between 0.0 and 4.0 $^{\text{px}}/_{\text{frame}}$ with a step of 0.1 $^{\text{px}}/_{\text{frame}}$ and convert them to the intensity domain using inverse FFT. Later, we truncate the kernels to remove small weights and normalize them. The resulting filters of size 1×11 are stored in a texture and used later in the rendering. Fig. 3 (a) shows examples of G_v filters in the intensity domain, which have the typical shape of sharpening filters. When they are applied to the high-resolution image, we orient them according to the local motion direction. While it is possible to precompute the filters for larger speeds, we found that for speeds exceeding 4 $^{\text{px}}/_{\text{frame}}$, the effect of resolution enhancement cannot be appreciated due to the large motion.

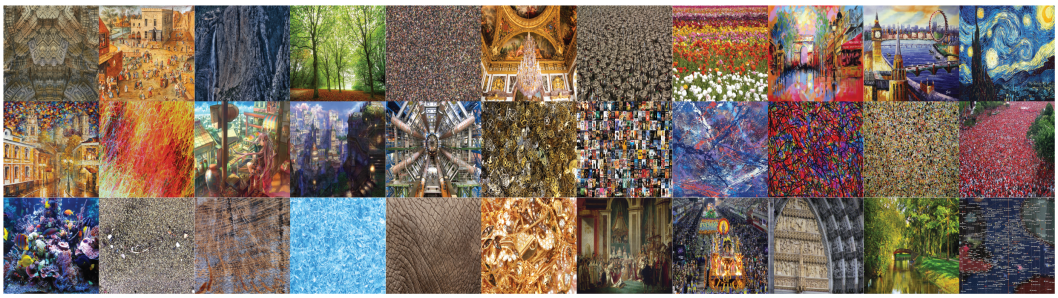

Fig. 4. A set of high-resolution images used for computing kernels $D_{v,k}$. Image courtesy: Charles F. Lindgren, asuscreative, Kevin Dooley, CERN, Daniel Proulx, bgfons.com, T.Voekler

4.2 Derivation of $D_{v,k}$ Filters

The goal of filters $D_{v,k}$ is replacing the decomposition of $\widetilde{I_H}$ into a set of sub-frames $\{I'_k\}$ (Eq. 9). We follow the idea presented in [Damera-Venkata and Chang 2009], where the authors considered a problem of decomposing a high-resolution image into multiple images displayed using several projectors (Sec. 2). In that work, the authors replaced an expensive optimization with a set of optimized filters. Since both problems lead to a similar formulation, we apply a similar procedure.

Finding filters $D_{v,k}$ which approximate Eq. 9 can be formulated as an optimization problem:

$$D_{v,k} = \underset{D_{v,k}}{\operatorname{argmin}} \left| T(D_{v,k} * I_H) - I_{OPT_k} \right|, \tag{11}$$

where I_{OPT_k} is an optimal sub-frame generated using the expensive optimization we want to replace with a filtering step. The solution to this optimization depends on I_{OPT_k}. This is because the filtering tries to approximate a constraint optimization (Eq. 9), i.e., the values of the optimized sub-frames need to lie in the range of $(0, 1)$, which gives more freedom for low contrast images that are less likely to suffer from invalidating the constraint. Therefore, to solve the above optimization and compute filters that generalize well for a wide range of different images/textures, we first collected a set of high-resolution images (Fig. 4). Then, we sample the range of possible velocities v between 0.0 and $4.0\,{}^{\text{px}}/_{\text{frame}}$ with a step of $0.1\,{}^{\text{px}}/_{\text{frame}}$. Since $D_{v,k}$ is a 2D filter which depends on velocity direction, we sampled both vertical and horizontal components of the velocity vector. Later, for each high-resolution image in our set and each velocity v, we perform an optimization proposed in [Didyk et al. 2010] to recover k sub-frames. We treat each of the sub-frames as the optimal image I_{OPT_k}. Given the optimal sub-frames, we solve optimization from Eq. 11 to recover corresponding filters $D_{v,k}$. The formulation leads to a system of linear equations which we solve in a least-square sense using a standard linear solver provided by Matlab. Since we want our filters to generalize across different images, we average filters across different high-resolution images. Examples of optimized sub-frame kernels for different velocities are illustrated in Fig. 3 (b). We limited the kernels to size 11×11 and stored them in a texture for later rendering. Due to symmetries in the filters, we only considered velocities in half of one quadrant (Fig. 5). This can significantly improve the storage requirements.

In accompanying supplemental material, we provide all the filters (G_v^{-1} and $D_{v,k}$) that were obtained in the above process.

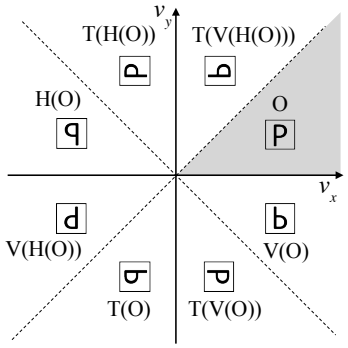

Fig. 5. The plot presents the space of the velocities. All kernels can be obtained through transformations on the kernel O in the gray area. Transformations T, V, and H represent transposition, vertical flip, and horizontal flip, respectively. We express kernels in other quadrant using kernels in the shaded region.

Fig. 6. Sub-frame images and their simulated result when a Gaussian filter is applied (top) and not applied (bottom). The additional filtering efficiently removes aliasing but does not destroy the effect of resolution enhancement.

4.3 Application

So far, we have described the derivation of the filters that replace optimization-based apparent resolution enhancement. To apply our technique to a real-time rendering, we synthesize a high-resolution image of a 3D scene once per k sub-frames. Since our technique is not able to fully reproduce the resolution of the input image, the image has to be prefiltered to avoid possible aliasing. To this end, we apply a small Gaussian filter with experimentally chosen parameter $\sigma = 1$. This step does not lower the resolution of the resulting image, but helps prevent visible aliasing (Fig. 6). During the rendering, we compute motion flow by taking the difference in positions for every vertex in two consecutive frames and rasterizing it into a buffer. The motion flow includes not only movement of objects, but also per-pixel optical flow resulting from head movements. This is essential, as it extends the benefit of our method to situations where objects are static, but there is a head movement. Next, for every pixel in the image, we examine the magnitude and direction of the optical flow and find the appropriate filters G_v^{-1} and $D_{v,k}$ in our precomputed database. Since we store the filters for discrete velocities, we use the filters which correspond to the closest velocity. We then use Eq. 9 to compute the sub-frames. All the steps are listed in Algorithm 1.

4.4 Discussion

As described above, the filter G_v^{-1} is applied to the entire image I_H once per k sub-images. $D_{v,k}$, on the other hand, is evaluated for every sub-frame, but it is computed only for the final pixels and not all the pixels in I_H. This means that although $D_{v,k}$ is larger than G_v^{-1} and more expensive to apply, it has to be evaluated for a smaller number of pixels, i.e., only those which are displayed on the screen. One can consider combining both filters in Eq. 9; however, this would result in much larger filters, which would increase the computation cost.

5 RESULTS

Our method was implemented using OpenGL. All the filtering was performed in a fragment shader based on the high-resolution input image, motion flow, and precomputed filters stored in textures. Furthermore, we considered 120 Hz displays as our testing environment. Similarly to [Didyk et al. 2010; Templin et al. 2011], we decomposed high-resolution images into three sub-frames. This also

Algorithm 1 Real-time resolution enhancement

```
subframeIndex ← 0
repeat
    if subframeIndex == 0 then
        Animate a 3D scene
        Render the high-resolution image
        Calculate optical flow
        Apply a Gaussian blur (σ = 1)
        Apply G_v^{-1}
    Apply D_{v,subframeIndex}
    Display sub-frame image
    subframeIndex ← (subframeIndex + 1) mod k
until Stopping condition
```

allows us to store filters $D_{v,k}$ in RGB channels, where each channel corresponded to filters for one of the three sub-frames. Consequently, we also rendered high-resolution input images in three times higher resolution than the resolution of the display.

5.1 Performance

To evaluate the performance of our technique, we considered rendering for off-the-shelf VR headset displays. As an example, we took the Oculus Rift display, which has a resolution of 1080×1200 per eye. Since the performance of our technique depends on the motion in the scene, e.g., static regions do not require our method, we considered a situation where every pixel undergoes our filtering. Therefore, the timings below present the worst-case scenarios. We measured the performance using an Nvidia Geforce GTX 980 and unoptimized implementation of our filtering.

Applying our technique to a full 1080×1200 display takes 8.9 ms. This includes 1.7 ms for Gaussian filtering , 1.2 ms for G_v^{-1}, and 6 ms for applying $D_{v,k}$. Note that the two first filters are applied only every three frames; however, we include them in the timing to provide a conservative upper bound for every frame. Although the above performance is already close to enabling support of a 120 Hz display, it still takes too long, considering that there is only 8 ms budgeted for rendering and our technique has to address two displays. Fortunately, our technique does not need to be applied on the entire screen; instead, we can exploit available eye-tracking technology, such as Pupil Labs' solution, to apply our technique only in the foveal region. According to recent work on foveated rendering [Patney et al. 2016], it is sufficient for current VR headsets to provide the highest resolution only for approximately 30 degrees of visual field. This means that for such headsets as Oculus Rift, applying our technique to $1/9^{th}$ of the screen will provide the desired quality. Applying our technique to such a portion of the screen takes 2.15 ms (4.3 ms for two displays), which includes 0.45 ms for Gaussian filtering, 0.6 ms for G_v^{-1}, and 1.1 ms for applying $D_{v,k}$. Although we did not test it, we believe that the portion of the screen could be further reduced. 30 degrees of foveal region was estimated for the native resolution of the screen, and since our technique provides higher resolution, the foveal region could potentially be smaller.

5.2 Results Simulation

Since the high-resolution result of our technique is achieved due to the temporal integration of the HVS, we present simulated results. We considered seven different test examples. Two of them consist of square images moving with different velocities (Fig. 7, left). These are taken from the previous

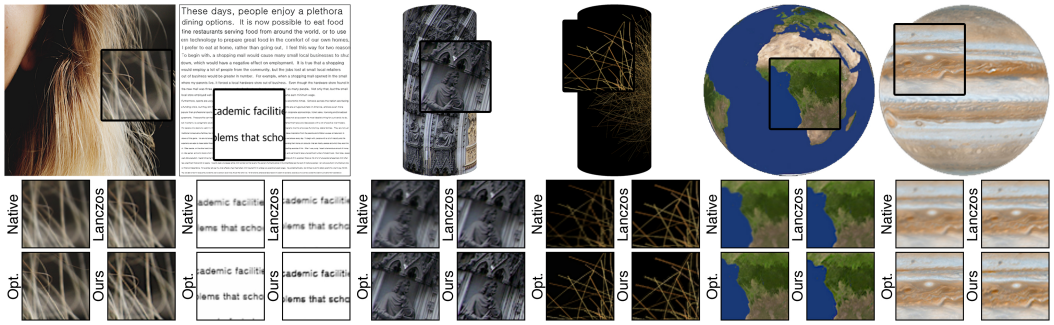

Fig. 7. Simulated results of our method with comparison to different techniques. For the two left-most images a linear motion was assumed, while the two cylinders and the spheres were rotating around a vertical axis. Image courtesy (in order 3rd, 5th, and 6th from the left): T.Voekler, James Hastings-Trew, James Hastings-Trew.

work [Didyk et al. 2010; Templin et al. 2011]. We also include two differently textured rotating cylinders (Fig. 7, middle) and two spheres (Fig. 7, right), as well as a complex 3D scene in which users can freely navigate (Fig. 1). The simulations present resolution enhancement capabilities for various spatially-varying speeds.

To demonstrate the effectiveness of our method, we compared our technique to three alternative solutions. The first one consists of native OpenGL rendering in the display resolution with bilinear texture interpolation. The second method uses the same high-resolution input as our technique, but it performs a standard downsampling of the image to the resolution of the display using Lanczos filtering. The last method is the optimization approach proposed in [Didyk et al. 2010; Templin et al. 2011].

The results of the comparison are presented in Fig. 1 and 7. While the full-size images present the scenes, not the results, the insets provide simulations of different methods for particular locations. It can be observed that the native rendering provides the worst quality. Lanczos filtering improves the resolution significantly, while our technique provides the best resolution consistently. When compared to the original optimization, our technique provides slightly worse results. This is mostly because the optimization approach can locally adapt to the content, while our filters are not content-dependent. However, in contrast to our technique, the optimization results cannot be obtained in real time.

6 USER EXPERIMENT

We conducted a user experiment to evaluate the quality of obtained images when observed by viewers. The participants were asked to compare the quality of the images produced by native rendering, Lanczos downsampling, and our technique.

Stimuli. The stimuli consisted of the scenes used in Sec. 5.2. Table 1 presents a summary of the stimuli with corresponding IDs used later in the analysis.

Equipment. The experiment was performed on two display setups. The first consisted of a desktop 30-inch Acer Predator Z1 monitor operating at 120 Hz and 2560×1080 resolution. The participants sat approximately 60 cm from the display. No strict viewing distance was enforced, and the participants could freely move their head while sitting in an upright position. The experiment was conducted using standard office lighting. The second display setup consisted of an Oculus Rift VR headset operating at 90 Hz with 1080×1200 resolution for each eye. To maintain a stable

Table 1. Description of the scenes used in the evaluation experiment.

#	Description	Figure
1	Orange colored painting (image)	Fig. 1, left cube
2	Blonde hair (image)	Fig. 7, left
3	Text (image)	Fig. 7, left
4	Cologne cathedral door (cylinder)	Fig. 7, middle
5	Artificial hair ball (cylinder)	Fig. 7, middle
6	Earth (sphere)	Fig. 7, right
7	Jupiter (sphere)	Fig. 7, right
8	3D scene	Fig. 1

framerate, we limited the size of the stimuli to 540×600 px. The stimuli were shown always in the center, and the outside region was filled with black.

Participants. We invited 15 unpaid participants for our experiment. All of them have normal or corrected-to-normal vision. Two of them were unable to conduct the experiment for the Oculus headset, due to their prescription glasses. The participants were not aware of the purpose of our technique.

Task. At each trial, a participant was shown a scene in three versions. Only one version of the rendering was presented on the screen at a time, and the participant could toggle between all three versions using defined keyboard keys. In the first part of the experiment, the subject was asked to choose the method which exhibits higher visual quality and confirm their choice using the keyboard. At this point, no additional explanation was provided. In the second part of the experiment, the sequence of stimuli was repeated, but in this round, the participants were directly asked to indicate a version of the scene which provided higher spatial resolution. The participants were given an unlimited amount of time to complete the experiment. At the beginning of each part, each participant received a written description of the task and keyboard keys used for answering the questions. At the end of the experiments, the participants were asked to indicate what was the main factor when judging the quality. They could choose between blur, flickering, and aliasing. Since aliasing is not a common term, we provided participants with a simple figure visualizing the problem.

Results. The results of the experiment are presented in Fig. 8. In all cases, our technique was preferred over Lanczos filtering and native rendering. Regarding the question about the overall quality, the difference is less prominent for the VR headset. This is mostly because at 90 Hz remaining flickering can be observed. Also for the 120 Hz, the quality scores of our technique are slightly lower than resolution judgments. This can be caused by remaining temporal fluctuations spotted by some participants, which would be in agreement with an observation from previous work [Berthouzoz and Fattal 2012a; Didyk et al. 2010] that report remaining flickering for 120 Hz. However, as it can be observed in our results, this problem is minor when it comes to the preference. In both cases, to solve the problem of remaining flickering, it is possible to apply a technique proposed by Didyk et al. [2010] which provides a trade-off between resolution enhancement and flickering. Also, increasing the parameter σ in our prefiltering step would lead to a similar effect. The importance of flickering and its influence on the overall quality judgment is further supported by the fact that a large part of participants reported flickering as an essential factor in the quality judgment (Fig. 9). We performed a statistical analysis to check whether the improvements of the quality and

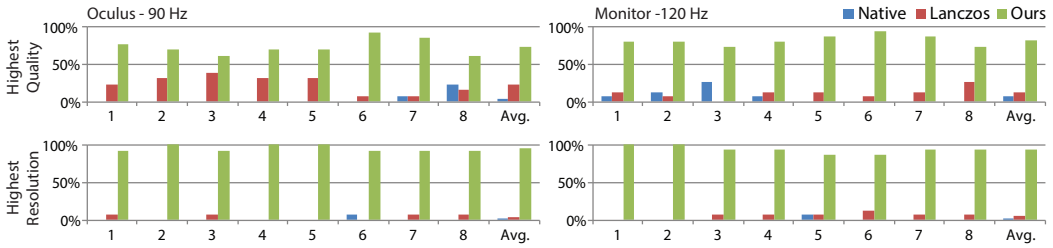

Fig. 8. The ratio of participants selecting the result of native rendering, Lanczos resampling and our technique when they are asked to pick the one with highest quality (top row) and highest resolution (bottom row). The numbers on the x-axis correspond to the description of the scenes in Sec. 6. We also provide results averaged across the scenes.

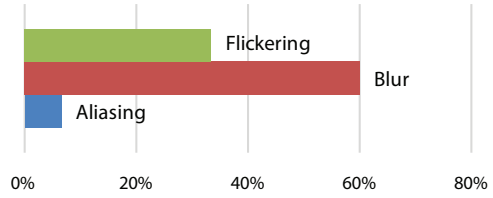

Fig. 9. The ratio of participants selecting flickering, blur and aliasing as the most important aspect of their quality judgment.

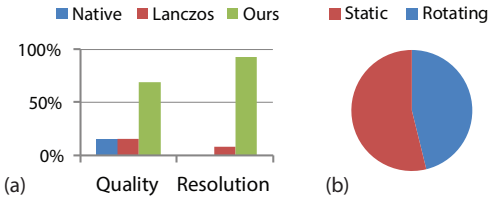

Fig. 10. The results of our experiment with artificial camera motion: (a) quality and resolution comparison; (b) the ratio of participants choosing the static and the rotating scene as the one without motion.

resolution provided by our technique (Fig. 8) are significant with respect to the native rendering and Lanczos filtering. Binomial test with Bonferroni correction revealed that the quality and resolution judgments for both VR headset and monitor are significantly higher when our technique is applied ($p<0.001$).

6.1 Artificial motion

As presented above, our technique can exploit both the motion of the objects in the scene as well as head motion of a user. One of the significant disadvantages of our method is that it cannot enhance resolution for the cases where neither the content nor the observer moves. Here, we investigate whether it is possible to introduce a small movement to the entire image presented to the observer such that the motion remains unnoticed, but our technique provides a significant resolution enhancement. Since the motion should be small enough to make the observer not notice it, we decided to apply a circular motion of a camera with an average on-screen speed of $0.4\,{}^{\mathrm{px}}/_{\mathrm{frame}}$. For the experiment, we used our Oculus VR headset with the scene from Figure 1. First, to investigate participants' awareness of the rotation, we allowed the participants to toggle between the static scene which had no camera rotation and the experimental rotating scene with the artificial motion. We asked participants to chose which of the two cases corresponded to a scene with subtle motion.

The results (Fig. 10, b) suggest that people had difficulties perceiving the subtle motion. To test whether the introduced motion led to any resolution enhancement, we again compared our technique to Lanczos filtering and native rendering. We followed the procedure from our main experiment and first asked about the overall quality, and later about the spatial resolution. The results of this experiment (Fig. 10, a) suggest that indeed the subtle motion led to resolution

enhancement. This demonstrates that it is also possible to apply our technique, although it was primarily developed for dynamic content, to static regions of the image and enhance resolution everywhere.

7 LIMITATIONS AND FUTURE WORK

The main limitation of the technique is the fact that it is designed to enhance resolution only in locations with local motion. We conducted a preliminary experiment which demonstrates that one can introduce an additional camera movement to improve resolution everywhere. This is done by introducing a micro-motion to the entire scene. While the motion is too small to be perceived, the HVS compensates for it, which triggers the mechanism required for our technique, i.e., sufficient motion for resolution enhancement. Although the results are promising, such a method requires more investigation regarding unwanted effects such as visual discomfort. Another limitation is that the resolution enhancement is directional, i.e., when an object is moving horizontally or vertically, only horizontal or vertical resolution can be enhanced. However, when the object's velocity contains both horizontal and vertical components, or a global camera motion is introduced, both horizontal and vertical resolution can be improved.

An essential requirement for our technique is a high-framerate screen. While 120 Hz desktop screens are already widely available, current headsets usually operate at 90 Hz. In such conditions, remaining flickering might be still a problem. However, we expect more higher-framerate VR displays in the future since they also are a prerequisite for a comfortable VR experience. Our technique will benefit from such developments.

Our technique exploits the fact that a viewer follows moving objects. To compute the perceived image, we use motion flow available during the rendering process and an assumption about linear motion. This allows us to correctly handle objects whose trajectory can be locally approximated by a straight line. Since our sub-frame generation accounts for movement in the scene, it can also be seen as a simple temporal extrapolation which reduces a temporal lag caused by the decreased framerate of high-resolution images. In cases where the assumption about linear motion does not hold, our technique may provide inaccurate results. However, we have not observed problems in practice.

In our derivation of filters G_v^{-1}, we assumed that the image is continuously displayed on a screen. For displays which utilize strobing backlight, it would be interesting to include this information into the kernel computation. In such case, the box filter that models the temporal integration performed by the human visual system would need to account for the lower persistence, for example, by reducing its spatial support.

Another avenue for future work is improving the efficiency of our technique. Since our method requires evaluating two convolutions with relatively small kernels, we believe that its efficiency can be further enhanced. The method could also be implemented in hardware and applied at the end of any rendering pipeline. Currently, the σ parameter for the prefiltering step of the high-resolution image is chosen experimentally. A better frequency analysis of our method could lower the filter size and improve the results. Also, our current filters estimation tries to find a compromise by providing filters that perform well for a wide range of images. We believe that the resolution enhancement capabilities can be improved with content-dependent filters.

8 CONCLUSION

We proposed a technique for real-time apparent resolution enhancement. The key feature of the technique is that, apart from a high-framerate display, it does not require any other specialized display or hardware modification. Comparing to similar methods, we replaced a computationally

expensive optimization process with two simple filtering steps which provide a significant speed-up. As a result, our technique can be easily applied to head-mounted displays with eye-tracking technology. The results of our method are demonstrated in simulation and a user experiment, which prove the benefit of our technique. Despite the limitations and possibilities for further improvements, our study demonstrates that our technique is already beneficial in the current form as it increases both overall quality and apparent resolution. We also provided preliminary results showing that such a technology has a potential for improving the resolution not only for moving content, as demonstrated in previous works, but also for all regions in the image. This would make our technique universal and the resolution enhancement independent of the image content. Furthermore, the results as well as the low computational cost of our technique suggest that, in the future, the apparent resolution enhancement could be a part of any rendering pipeline for head-mounted screens.

REFERENCES

Will Allen and Robert Ulichney. 2005. 47.4: Invited Paper: Wobulation: Doubling the addressed resolution of projection displays. *SID Symposium Digest of Technical Papers* 36, 1 (2005), 1514–1517.

Floraine Berthouzoz and Raanan Fattal. 2012a. Apparent resolution enhancement for motion videos. In *Proceedings of the ACM Symposium on Applied Perception*. ACM, 91–98.

Floraine Berthouzoz and Raanan Fattal. 2012b. Resolution enhancement by vibrating displays. *ACM Transactions on Graphics (TOG)* 31, 2 (2012), 15.

Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. 1990. Human photoreceptor topography. *Journal of Comparative Neurology* 292, 4 (1990), 497–523.

Niranjan Damera-Venkata and Nelson L Chang. 2009. Display supersampling. *ACM Transactions on Graphics (TOG)* 28, 1 (2009), 9.

Piotr Didyk, Elmar Eisemann, Tobias Ritschel, Karol Myszkowski, and Hans-Peter Seidel. 2010. Apparent display resolution enhancement for moving images. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 113.

Thomas Engelhardt, Thorsten-Walther Schmidt, Jan Kautz, and Carsten Dachsbacher. 2014. Low-cost subpixel rendering for diverse displays. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 199–209.

Felix Heide, Douglas Lanman, Dikpal Reddy, Jan Kautz, Kari Pulli, and David Luebke. 2014. Cascaded displays: Spatiotemporal superresolution using offset pixel layers. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 60.

Benjamin Kading and Jeremy Straub. 2015. Enhancing head and helmet-mounted displays using a virtual pixel technology. In *SPIE Defense and Security*. International Society for Optics and Photonics, 947011–947011.

Michael Kalloniatis and Charles Luu. 2007. Temporal resolution. *Webvision: The Organization of the Retina and Visual System* (2007).

Michiel A Klompenhouwer and Leo Jan Velthoven. 2004. Motion blur reduction for liquid crystal displays: Motion-compensated inverse filtering. In *Proceedings of SPIE*, Vol. 5308. 690–699.

Belen Masia, Gordon Wetzstein, Piotr Didyk, and Diego Gutierrez. 2013. A survey on computational displays: Pushing the boundaries of optics, computation, and perception. *Computers & Graphics* 37, 8 (2013), 1012–1038.

Dean S Messing and Louis J Kerofsky. 2006. Using optimal rendering to visually mask defective subpixels. In *Proceedings of SPIE*, Vol. 6057. 236–247.

Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards foveated rendering for gaze-tracked virtual reality. *ACM Trans Graph (Proc SIGGRAPH Asia)* 35, 6 (2016), 179.

John C Platt. 2000. Optimal filtering for patterned displays. *IEEE Signal Processing Letters* 7, 7 (2000), 179–181.

Road to VR. 2017. Understanding pixel density & retinal resolution, and why it's important for AR/VR headsets. (2017). https://www.roadtovr.com/

Michael Stengel, Martin Eisemann, Stephan Wenger, Benjamin Hell, and Marcus Magnor. 2013. Optimizing apparent display resolution enhancement for arbitrary videos. *IEEE Transactions on Image Processing* 22, 9 (2013), 3604–3613.

Krzysztof Templin, Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. 2011. Apparent resolution enhancement for animations. In *Proceedings of the 27th Spring Conference on Computer Graphics*. ACM, 57–64.