Apparent Resolution Enhancement for Motion Videos

Floraine Berthouzoz
University of California, Berkeley

Raanan Fattal
Hebrew University of Jerusalem

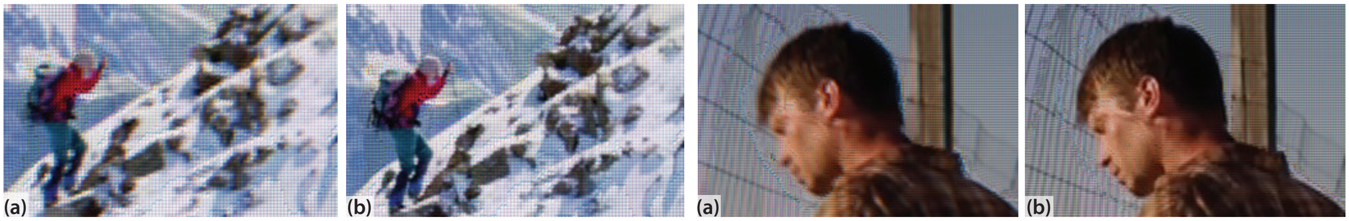


Figure 1: *A comparison between frames a viewer perceives when watching the input video (a), and the ones produced by our method (b). Image credits: Michael Fisher*

Abstract

In this work we increase the apparent resolution of videos when viewed on a high-refresh rate display by making use of perceptual properties of the visual system. We achieve this enhancement by exploiting the viewer's natural tendency to track moving objects in videos which causes the screen pixels to be projected at different sub-pixel offsets onto the retina. We estimate the eye motion using optical flow and use it to compute multiple low-resolution frames for each input frame. By watching these new frames at a high frame-rate, the viewer's eyes integrate them over time and merges them into a single perceived frame with a denser pixel layout. In this work we also advance the existing approaches for resolution enhancement in the following ways. We combine current display resolution enhancement with super-resolution methods to enhance input videos that are at the display resolution. We derive a new perceived video model that accounts for actual camera sensor and display pixel shapes in order to achieve optimal enhancement. We analyze the degeneracies that certain motion velocities introduce to super-resolution and resolution enhancement, and offer algorithmic solutions for handling these scenarios as well as other difficulties that arise when dealing with the optical flow of natural videos.

A user study finds that our approach achieves a noticeable increase in the apparent resolution for videos even when viewed on regular hardware (60Hz), and further enhances resolution when viewed on higher refresh rate displays (120Hz).

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—[Display Algorithms]

Keywords: resolution enhancement, image processing, computational photography

1 Introduction

The recording and sharing of videos have become increasingly easy to a point where any person can record and upload a video from his/her cell phone within a few minutes. As a consequence, videos have become omnipresent in our lives. Since most people use a variety of personal devices (camera recorder, cell phone, iPad, etc.) to record and view videos, it cannot be expected that the video will be viewed on a device with the same resolution as the recording device. Often higher resolution videos have to be downscaled to fit the resolution of the viewing device and thus the fine detail of the original scene is lost.

Increasing the resolution of displays is challenging, because more fully functional pixel units must be aligned within a smaller area. For this reason, even modern LCD displays only reach two mega-pixels and fall below nowadays consumer camera sensors which exceed ten mega-pixels. Furthermore, even if displays were of higher resolution, lower resolution videos might be more desirable because of their low transmission bandwidth and storage requirements that make them ideal for fast and easy sharing.

In view of these limitations Didyk et al. [2010a] recently developed an alternative perception-based method for increasing the perceived resolution of displays. They exploit the retinal integration property of the eye to increase the apparent resolution of still images by moving them across the screen. When tracking the moving image, the viewer's eye integrates light at different sub-pixel offsets along the motion trajectory. Didyk et al. rapidly display different images at these sub-pixel offsets that are integrated by the retina into a single image. The displayed images are designed to produce a perceived image with an apparent resolution that is higher than the display resolution. Templin et al. [2011] extend this approach to animations by relying on the natural tendency of a viewer to track moving objects in the video. This tracking leads to the sub-pixel offsets needed for retinal resolution enhancement. They predict the path of the eye tracking via optical flow calculations and then use it to compute the low-resolution frames (LRFs) that optimally convey the high-resolution input frames when viewed on a high-refresh rate display. They demonstrate their approach on high-resolution and high frame-rate computer-generated videos with smooth motion fields.

In this paper we extend this approach in several respects. We show that the display resolution enhancement formulation is directly related to multi-frame video super-resolution [Patti et al. 2002; Elad and Feuer 2002; Borman and Stevenson 2002], where high-resolution video frames are extracted from multiple low-resolution frames. We exploit this formal connection and combine the two approaches to achieve a solution that increases *both* the resolution of the input video as well as the device displaying it. We can thus generate new LRFs that will be perceived as higher resolution on a high-refresh rate display even if the input video is at the display resolution. The integration of these two methodologies results in a more efficient solution than executing them separately and allows us to introduce a consistent normalization that handles degeneracies in the optical flow.

Additional contributions of this work are: (i) a principled treatment for degenerate motions as well as discontinuous optical flow fields, which are two abundant phenomena in natural video sequences. These algorithmic modifications are based on an analysis of the inherent limitations in resolution enhancement and estimating the effect that different flow vectors have on our system. (ii) Our model accounts for the actual shape of the camera sensors and display pixels and introduces matrix conditioning and kernel normalization steps in order to cope with various degeneracies that arise in such a system. Finally, (iii) our method computes the optimal set of LRFs for each input frame *separately* and thus applies for standard 30 frames-per-second input video sequences videos of arbitrary length.

Figure 1 compares an input frame with the frame computed using our method. Our perceived frame contains more high-frequency detail for the man's face and the mountaineer's clothes. We show that in ideal settings, our method can achieve a resolution enhancement of up to a factor two for such moving image areas when compared to the screen's native resolution. A user study comparing our videos to the input videos and to naively sharpened videos, finds that on average 71% of the subjects perceive our video to have the highest resolution when viewing them at 60Hz. This number increases to 88% of the subjects when the videos are played at 120Hz.

2 Related Work

Our method builds on several other areas of related work.

Perception-Based Techniques. Several existing display systems make use of the retinal integration property of the eye. For example, video interlacing used in CRT television sets is a technique that alternates between displaying the even and odd horizonal video scanlines of the screen. At every instant, only half of the pixels are shown, but the viewer integrates this information temporally and perceives one continuous image. As we discussed in the Introduction, recent works in computer graphics also use the temporal and spatial integration properties of the eye to increase the resolution of displays spatially [Templin et al. 2011; Didyk et al. 2010a] as well as the temporal resolution of computer-generated videos [Didyk et al. 2010b]. Berthouzoz and Fattal [2012] vibrate a display by a very small amplitude to obtain the sub-pixel offsets between several rapidly displayed low-resolution images. The viewer then merges these images into a single perceived image of higher resolution.

Display Resolution Enhancement Techniques. Subpixel rendering is a technique that increases display resolution by considering the red, green and blue color channels as separate pixels [Platt 2002]. This technique enhances only the resolution of the luminance image component and is limited to the horizontal direction. Researchers have also developed several techniques for projectors that are known to suffer from insufficient resolution. Their technique consists of either projecting several low-resolution images at small offsets [Damera-Venkata and Chang 2007; Jaynes and Ramakrishnan 2003] or deviating the light coming from the project while displaying low-resolution images [Allen and Ulichney 2005]. Our work is similar to these projector techniques in the sense that we generate an optimal set of LRFs for each input image. But in our case, the target device is a single display. Our method therefore does not have to account for intra- and inter-projection variations that require careful geometric, photometric and color calibrations.

Multi-frame Super-Resolution. Super-resolution techniques aim to reconstruct a high-resolution image from a set of low-resolution input images taken at different translational offsets. Super-resolution has been applied to sets of images taken from the same static scene [Irani and Peleg 1990; Park et al. 2003], as well as low-resolution videos [Patti et al. 2002; Elad and Feuer 2002; Borman and Stevenson 2002]. In videos, optical flow is typically used to estimate the translational offset between each frame. Ben-Ezra et al. [2004] avoid using optical flow by physically shifting the camera sensor as it records the video. Our work combines the model used in super-resolution with our display resolution enhancement approach, which allows us to generate a set of enhanced LRFs from a low-resolution input video.

3 Method

We enhance the apparent resolution of a video when viewed on a low resolution display by replacing every input frame by several lower resolution frames that, when combined together, give rise to a higher perceived resolution. In order to compute this set of LRFs we have to model the image that is perceived when the eye tracks an object in the scene. We start by formulating a static display (and hence a static gaze) and then extend it to motion videos. The function that describes the appearance of a single static image L in continuum is given by

$$V_{static}(y) = \sum_{x=1}^{N} P_{dis}(y-x)L(x), \tag{1}$$

where x is an integer pixel index, N is the number of pixels in L, y is a real-valued point coordinate on the screen plane, and P_{dis} is the point spread function (PSF) of the display, i.e., a function that describes the shape of a single-pixel turned on, as it appears on the display.

When displaying several LRFs within a period that *falls below* the retinal integration time, the viewer fuses the images and perceives them as one single image

$$V_{static}(y) = \sum_{x=1}^{N} \int_{0}^{T} P_{dis}(y-x)L_t(x)dt, \tag{2}$$

where T is below the retinal integration time. Assuming the eye is following a moving object, we need to model the relative motion between the screen and the tracked point. We parameterize the viewing coordinates of the apparent image by $y-\varphi(t)$, where $\varphi(t)$ describes the trajectory of the tracked point at time t.

$$V_{tracking}(y) = \sum_{x=1}^{N} \int_{0}^{T} P_{dis}(y-x-\varphi(t))L_t(x)dt. \tag{3}$$

In practice L_t is not a continuous function in time as we can only display a small finite number n of LRFs.

$$V_{tracking}(y) = \sum_{x=1}^{N} \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} P_{dis}(y-x-\varphi(t))L_i(x)dt, \tag{4}$$

where $[t_i, t_{i+1}]$ is the time interval during which the i-th LRF is displayed. This model accounts for eye movements that do not depend on the particular point in the frame. This is not the case for general videos that contain multiple objects moving in different directions and speeds. In such cases viewers typically scan the scene quickly and then track one object of interest in the frame [Boccignone et al. 2002; Henderson 2003]. Hence, similarly to [Templin et al. 2011; Boccignone et al. 2002], we predict the eye movement $\varphi(t,x)$ at every pixel x of the input frame as the local motion of the video content and estimate it by computing a dense optical flow [Brox

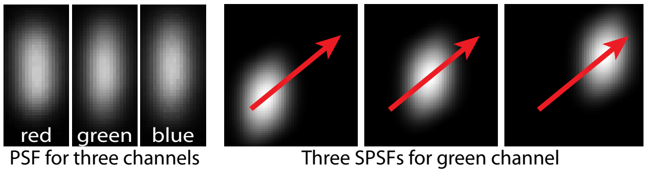


Figure 2: *(left) Static PSFs. (right) Three SPSFs generated for the green channel. We show in red the linear trajectory along which the PSF was smeared. All images were magnified by a factor 32.*

et al. 2004]. Thus, we replace $\varphi(t)$ in (4) by $\varphi(t,x)$.

$$\begin{aligned} V_{tracking}(y) &= \sum_{x=1}^{N}\sum_{i=1}^{n}\int_{t_i}^{t_{i+1}} P_{dis}(y-x-\varphi(t,x))L_i(x)dt \\ &= \sum_{x=1}^{N}\sum_{i=1}^{n} S_{dis,i}(y,x)L_i(x), \end{aligned} \tag{5}$$

$$\text{where } \; S_{dis,i}(y,x) = \int_{t_i}^{t_{i+1}} P_{dis}(y-x-\varphi(t,x))dt. \tag{6}$$

We call these time-averaged display point spread functions, $S_{dis,i}$, *smeared point spread functions* (SPSFs). In Figure 2 we show an example of a static PSF and its SPSFs. Assuming the display refreshes at uniform periods, the time intervals $[t_i,t_{i+1}]$ divide the time between the two input frames into n equal intervals. Thus, for each pixel, there will be n SPSFs (one for each of the n LRFs), that are smeared along $1/n$-th of the optical flow vector for that pixel.

Finally, we use the perceived image model (5) to compute the LRFs, L_i, that produce an optimal approximation for a given high-resolution input frame, $H(y)$, in the l_2 sense, by solving the following quadratic minimization

$$\min_{L_{1\ldots n}} \int \Big(\sum_{x=1}^{N}\sum_{i=1}^{n} S_{dis,i}(y,x)L_i(x) - H(y) \Big)^2 dy. \tag{7}$$

In practice, we solve this minimization problem on a discrete pixel grid, in which the y coordinates are discretized. The input image H and the SPSFs are specified at this grid. The grid resolution is greater or equal to the *target resolution*, which is defined as the maximum resolution enhancement that we can obtain with our method. The target resolution is determined by the number and offset between the LRFs. If we use n LRFs then, under ideal conditions with a uniform arrangement between the LRFs (as we discuss in Section 3.2), we will at most be able to match n times the high frequency content of the LRFs. The target resolution is therefore equal to the total number of pixels in the n LRFs.

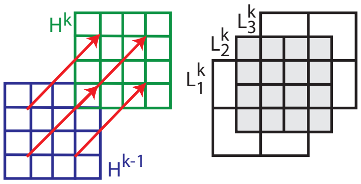
This discretization allows us to express the minimization in (7) in matrix form. We define a matrix $\mathbf{W}_i$ for each LRF L_i. Each $\mathbf{W}_i$ contains as its row vectors the SPSF $S_{dis,i}$ for every pixel x in L_i. By combining these matrices into one matrix $\mathbf{W} = [\mathbf{W}_1,\ldots,\mathbf{W}_n]$, the minimization (7) becomes

$$\min_{L} \left(\mathbf{W}L - H\right)^2, \tag{8}$$

where $L = [L_1,\ldots,L_n]^T$ is the column vector containing the unknowns of all the n LRFs that we are solving for, and H is the vector of the high-resolution input frame pixels. We solve (8) for each color channel (red, green, blue) independently. Note that Equation (8) optimizes for the set of LRFs that best approximate *each* high-resolution input frame, meaning that we solve for each input frame independently.

Discussion. Unlike the previous method of Templin et al. [2011], our perceived video model (5) allows us to use the precise display PSF function in order to achieve optimal performance. We demonstrate the importance of this consideration in Section 4. Furthermore, since we compute multiple LRFs from each input frame our method applies to regular 30Hz input videos. Finally, by processing each input frame individually our memory requirements do not depend on the video length.

We are able to achieve resolution enhancement for videos since the eye tracking causes each of the LRFs to be projected at a slightly offsetted location onto the retina. For example, the inset (left) shows two consecutive high resolution frames H^{k-1} and H^k, and the motion vector between these two frames. Even though the $n=3$ LRFs that approximate H^k are at half the resolution of H^k (right), the offset between these images gives rise to a higher resolution grid where the LRFs overlap (shaded gray area). We thus achieve resolution enhancement by mimicking a denser pixel layout. The spatial layout of the LRFs plays a critical role in terms of the amount of resolution enhancement achievable. We discuss this aspect in Section 3.2.



3.1 Combination with Super-Resolution

The method described so far uses a high-resolution input video. We can use multi-frame super-resolution techniques [Patti et al. 2002; Elad and Feuer 2002; Borman and Stevenson 2002] to reconstruct high-resolution frames from several low-resolution frames and extend our approach to operate on low-resolution input videos. Rather than using these methods as a pre-processing step, we propose to integrate our display resolution enhancement method and super-resolution into a single model.

In super-resolution the relation between the low-resolution frames $F_1,\ldots,F_m$ and the high-resolution frame H is modeled by

$$\begin{aligned} F_i(x) &= \int_{t_i}^{t_i+s}\int P_{cam}(x-y-\varphi(t,x))H(y)dydt \\ &= \int S_{cam,i}(x,y)H(y)dy, \end{aligned} \tag{9}$$

where P_{cam} is the camera PSF, $\varphi(t,x)$ describes the motion in the video between frame F_i and H, s denotes the period where the camera shutter is open and

$$S_{cam,i}(x,y) = \int_{t_i}^{t_i+s} P_{cam}(x-y-\varphi(t,x))dt. \tag{10}$$

Here the time intervals $[t_i,t_i+s]$ correspond to the period where the i-th frame was shot. Note that this model is very similar to (5) where the LRFs and the H exchanged roles. Indeed, super-resolution techniques and our resolution enhancement method have the opposite goal. While we compute a set of LRFs from a high-resolution input image, super-resolution techniques reconstruct a high-resolution image from a set of LRFs. Upon discretization of the y coordinate, Eqn. (9) can be written as

$$F = \mathbf{S}H, \tag{11}$$

where $\mathbf{S}$ is a matrix that contains as its column vectors the SPSFs for every pixel x. The vector $F = [F_1,\ldots,F_m]^T$ is the column vector containing the m low-resolution input frames, and H is the vector of unknown high-resolution pixels. Ordinarily, the discretization resolution of H must not be higher than the total number of constraints

Algorithm 1 Constrained Iterative Gauss-Seidel Solver

```
    Define   $\mathbf{A} \leftarrow \mathbf{SW}$ and $b \leftarrow F$
  $L^{(0)} \leftarrow b/Diag(\mathbf{A})$   (initialization)
  $L_i^* \leftarrow L_i^k - \beta \mathbf{A}_{ii}^{-1}(\sum_{j<i} \mathbf{A}_{ij} L_j^{k+1} + \sum_{j \geq i} \mathbf{A}_{ij} L_j^k - b_i)$ (update)
  $L_i^{(k+1)} = \begin{cases} L_i^* & \text{if } L_i^* \in [0,1] \quad \text{(clamping)} \\ 0 & \text{if } L_i^* < 0 \\ 1 & \text{if } L_i^* > 1 \end{cases}$
```

in (9) which is equal to the total number of pixels in the m LRFs. Finally, we compute the high-resolution frame H by minimizing

$$\min_H \left(\mathbf{S}H - F\right)^2. \tag{12}$$

To combine this model with our display resolution enhancement, we use the approximation $\mathbf{W}L \approx H$, which we optimize for in (7), and plug it into (12). We obtain

$$\min_L \left(\mathbf{SW}L - F\right)^2. \tag{13}$$

This minimization allows us to compute an optimal set of LRFs from a low-resolution input video, such that the perceived video appears as higher resolution. Here as well, we solve Eqn. (13) for each color channel independently. In our implementation, we either generate $n=2$ or $n=3$ LRFs, L_i^k, where k denotes the index of the input video frame. To compute $n=3$ LRFs $L = [L_1^k, L_2^k, L_3^k]^T$, we use $m=3$ input frames $F = [F^{k-1}, F^k, F^{k+1}]^T$. In general we always use $n=m$ and therefore for the case of $n=2$ only use the input frames $F = [F^{k-1}, F^k]$. We compute the optical flow from frame $k-1$ to frame k and from $k+1$ to k to estimate the eye tracking needed for $\mathbf{W}$ and the offsets needed in the super-resolution component $\mathbf{S}$.

The resolution of the input frames in F need to be greater or equal to the number of unknowns in the LRFs L. If these numbers are equal, i.e., the display and the input video have the same resolution, the matrix $\mathbf{SW}$ becomes square and regular and the optimal solution is given by $\mathbf{SW}L = F$. This system is better conditioned than the normal equation $(\mathbf{SW})^T\mathbf{SW}L = (\mathbf{SW})^T F$ which results from (13) upon differentiation (the latter consists of a matrix times itself and thus its condition number raises to the power of two). Thus, if the input frames' dimension is close to the LRFs', it may be beneficial to reduce their resolution.

Constrained Linear Solver. The matrix $\mathbf{SW}$, as well as $\mathbf{W}^T\mathbf{W}$ from Section 3, are highly sparse due to the limited overlap between the SPSFs and can be efficiently solved via iterative linear solvers. We use the Gauss-Seidel iteration with a small modification. Pixel values are confined to a limited range $[0,1]$, because an LCD cannot produce 'negative' light intensities and also has a maximal intensity it can emit. Therefore, we clamp the LRFs pixel values, $L_i(x)$, after every iteration by setting them to zero and one if they run below or above these values respectively. This operation corresponds to a *projections onto convex sets* scheme [Youla 1978] and is known to converge to the optimal solution. We provide pseudo-code for this procedure in Algorithm 1.

3.2 Matrix Conditioning and Normalization

The optical flow defines the SPSFs and hence equations (8) and (13) (through $\mathbf{W}$ and $\mathbf{S}$). The computed flow fields are often complex and introduce several difficulties in our construction. Certain flow vector magnitudes cause nearby kernels to overlap. This prevents resolution enhancement and super resolution and leads to degenerate linear systems. Such scenarios also happen due to spatial changes in the flow field. For example, at interfaces between kernels of a slowly and rapidly moving objects, the rapid kernels may overlap the slower ones. Moreover, the spatial variability in the arrangement and amount of stretching of the kernels leads to different intensity values in the computed LRFs produced by the *same* target high-resolution values. Since these cases can lead to visual artifacts in the resulting video such as the ones shown in Figure 3(bottom), it is crucial for our method to handle them and we describe our approach in this section.

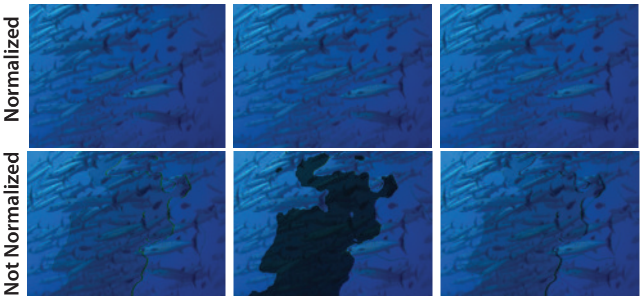
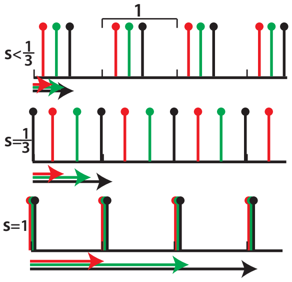


Figure 3: *Example LRFs generated with (top) and without (bottom) matrix conditioning and normalization.*



In the inset, we show the samples that mark the centers of the SPSFs (in 1D) for various motion vectors φ. The sample colors red, green and black correspond to three LRFs and the arrows indicate their offset along the motion vector φ. If we denote by s the offsets between the grid in each LRF in units equal to the grids' spacing, we see that integer s lead to overlapping grids (bottom) and when s is an integer plus $1/n$ then the n grids form a uniform grid with n times higher resolution (middle). In the Appendix we provide an analysis that estimates the effect s has on the condition number of $\mathbf{W}^T\mathbf{W}$ and $\mathbf{SW}$. This analysis predicts that the condition number of (8) and (13) have $O((s - \lfloor s \rfloor)^{-1})$ dependency on s. Large condition numbers imply that these systems produce large magnitude solutions despite the right-hand-side input data being bound. Such solutions are not feasible in our system since we clamp the LRFs values to $[0,1]$ in Algorithm 1, meaning we inherently cannot achieve resolution enhancement for the systems with overlapping kernels (seen in inset (bottom)). Therefore, we turn our enhancement mechanisms 'off' and modify the matrices to produce bounded solutions. In the case of super resolution we display the same low-resolution content in all the overlapping LRFs kernels and in the case of resolution enhancement we simply assign the same pixel values to the overlapping kernels. In both cases, these degeneracies and their correction is found and fixed quite easily. We inspect the off-diagonal elements of $\mathbf{W}^T\mathbf{W}$ or $\mathbf{SW}$, depending on the model used. These values equal to the dot-products between nearby kernels and therefore highly overlapping kernels produce values close to one (assuming the kernels are normalized in the l_2 sense). We map off-diagonal values which are greater than 0.9 through $f(x) = (1-x)/2$ which maps them to be close to zero. This reduces the coupling between variables in the linear system (making it closer to a diagonal matrix) and improves its condition. Note that the overlapping inferred from the off-diagonal elements is an indirect measure for the local offsets s. Expressing this correction step in terms of overlaps has the advantage to also handle singularities due to discontinuous vector fields.

Different flows produce different SPSFs with different arrange-

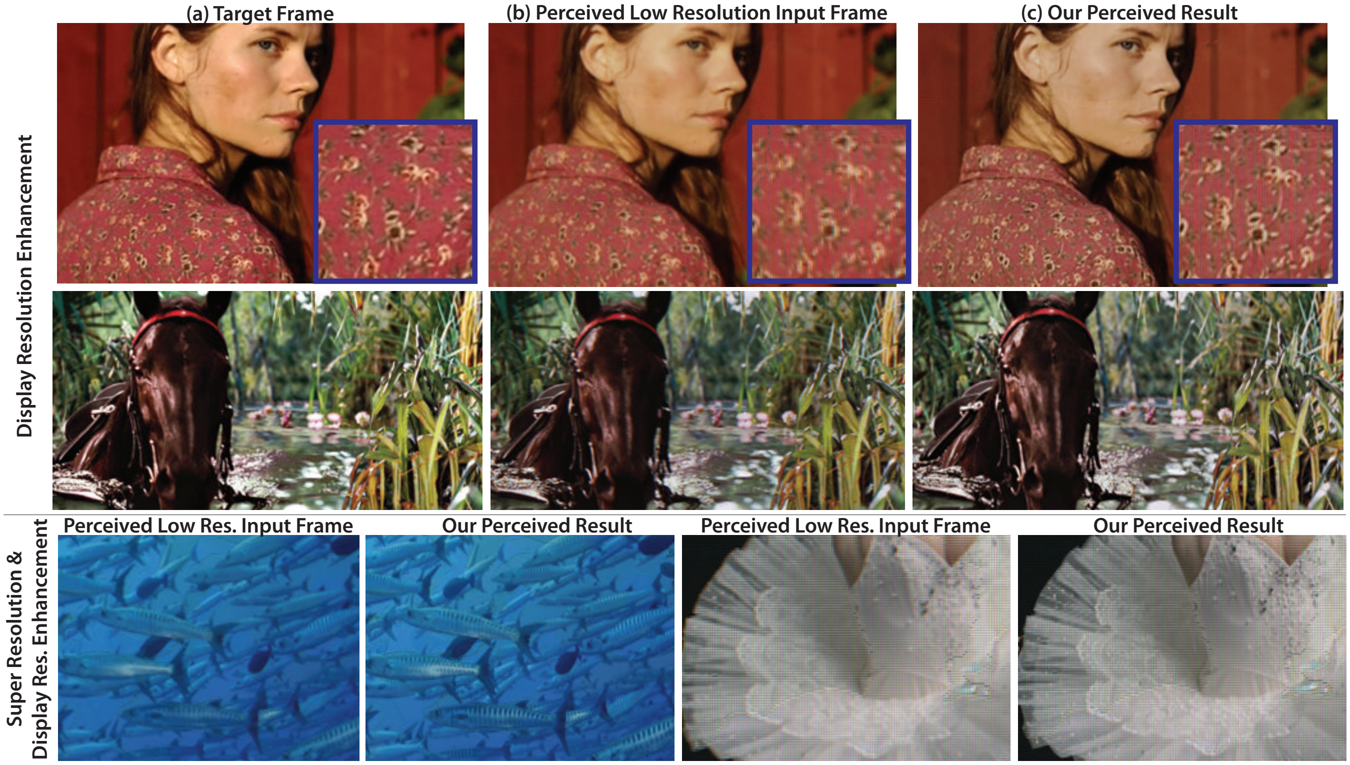

(a) Target Frame
(b) Perceived Low Resolution Input Frame
(c) Our Perceived Result
Display Resolution Enhancement
Super Resolution & Display Res. Enhancement
Perceived Low Res. Input Frame
Our Perceived Result
Perceived Low Res. Input Frame
Our Perceived Result

Figure 4: *(top) (a) Target frame at twice the display's resolution, (b) the perceived image when viewed on a lower resolution display, (c) the perceived frame using our method. (bottom) The perceived input frames at the display resolution and our perceived output frames. Image Credits: (upper row) Michael Fisher.*

ment. Therefore, there is no guarantee that constant right-hand-side input image will produce constant LRFs, implying that different intensity values can result from the same input value. Furthermore, discontinuities in the flow may lead to more severe under- and over-coverage of pixels by kernels. To prevent these artifacts, we add a normalization step where we solve for a phantom constant input image with pixel values set to 0.5, and use the true flow vectors extracted from the video to construct the matrix (e.g. **SW**). We then normalize the matrix by multiplying it from the right with a diagonal matrix containing the values of $L/0.5$ in its diagonal. This normalization ensures that constant input values will produce constant LRFs (Figure 3(top)).

3.3 Implementation

We tested our model on a 120Hz Samsung SyncMaster 2233RZ monitor. Here we describe various design choices and measurements that we made for this work.

Number of LRFs. Since the amount of resolution enhancement depends on the number of LRFs used, we would like n to be as high as possible. However, the LRFs also need to be displayed fast enough, so that the eye temporally integrates them into a single image. If we display the LRFs too slowly, the viewer might perceive flickering [Kalloniatis and Luu 2009]. Existing reports quote 40Hz as the safe frequency in terms of apparent flickering [Didyk et al. 2010a], so we opted for using $n=3$ LRFs at 40Hz or $n=2$ at 60Hz with our 120Hz display. Our user study (Section 5) shows that viewers still perceived very minor flickering at sharp edges. Didyk et al. [2010a] developed a post-processing step that eliminates flickering for $n=4$ LRFs at 30Hz, but similarly to using fewer LRFs, this post-processing step would also undermine the amount of resolution enhancement achieved.

PSF Acquisition. In order to obtain LRFs that are optimized for the display we use, we carefully acquire the static display PSF (P_{dis}) for each color channel by photographing the LCD panel as it displays a single red, green and blue pixel each time. We acquire these functions at a very high resolution using a macro magnifying lens. We then parametrize the PSF with gaussian kernels by searching for the parameters (μ, σ^2) that best approximate P_{dis} in l_2 norm. We generate the SPSFs, $S_{dis,i}$, for each pixel x by stretching the parametrized P_{dis} along the eye motion vector $\varphi(t,x)$. Specifically, we parametrize $S_{dis,i}$ by

$$e^{-x^T(\mathbf{RSR^T})^T\Sigma^{-1}(\mathbf{RSR^T})x} \tag{14}$$

where Σ^{-1} is a diagonal matrix with $1/\sigma^2$ in its diagonal elements, R is a rotation matrix that rotates the kernel in the direction of the trajectory $\varphi(t,x)$, and S stretches the kernel along this direction and is given by

$$S = \begin{pmatrix} 1 - \frac{||\varphi(t,x)||}{n} c & 0 \\ 0 & 1 \end{pmatrix}. \tag{15}$$

The amount of stretching is determined by $S_{1,1}$ and is proportional to the length of the motion vector $||\varphi(t,x)||/n$. We set $c=0.05$ which best approximates kernel's stretching along $1/n$-th of the trajectory. Figure 2 shows both the acquired static PSFs and the computed SPSFs. Similarly to existing reports [Park et al. 2003], we also approximate the camera PSFs, P_{cam}, using a Gaussian kernel and introduce motion blurring to the camera's SPSFs, $S_{cam,i}$, using (14) with $c=0.02$.

4 Results

To evaluate our model, we run our method on a collection of 30fps video clips of a total length of four minutes. Figure 1 and 4 show example results taken from these clips using $n=3$. The results shown in Figure 4 (top), were produced by the display

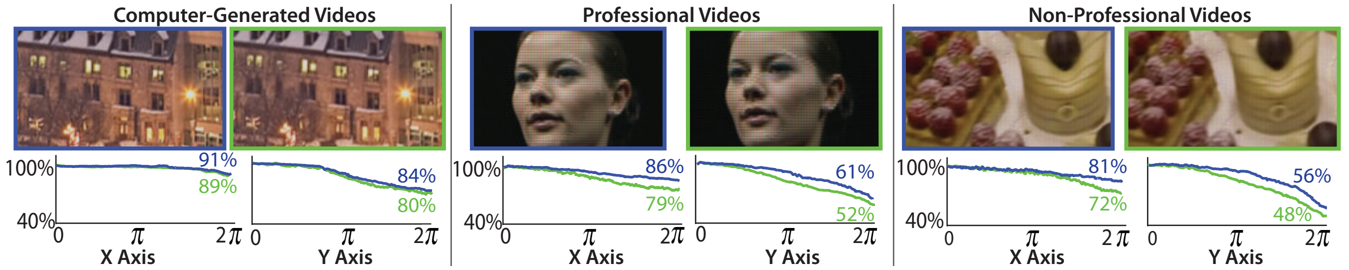


Figure 5: *For the three types of videos, we compute the percentage of magnitude spectrum that our results achieve compared to the target frame when using our combined method (green) versus our display enhancement method only (blue).*

resolution enhancement method described in Section 3 using a high-resolution input video. In Figure 1 and 4 (bottom), we show the results produced by combining our display resolution enhancement method with super-resolution (Section 3.1), given a low-resolution input video. The images shown here are the *perceived images* (i.e., $\mathbf{W}L$) which model how every part of the frame looks on the display when the viewer tracks it. Our results show more high-frequency detail than the low-resolution input videos or the input frames that were naively downscaled to the display resolution (i.e., filtering followed by subsampling) and compare well to the target images. For example, the patterns in the woman's skirt or the stripes on the fishes are better resolved in the frames computed with our approach. We perform additional tests to better evaluate our proposed approach. On the project website [1], we include several result videos that the reader can view on his/her own display as long as its refresh rate is set to 60Hz.

Spectral Analysis. To better quantify the enhancement we can achieve using our combined approach and the resolution enhancement approach only, we run both methods on the same set of high-resolution videos and downscale the input videos for the combined approach. We then compare the spectral content of our resulting perceived frames to that of the target frames by averaging it over twenty randomly selected frames. We perform this test for computer-generated videos that contain high frequency content and a known ideal flow field (as explained in 3.2), videos of professional quality, and videos taken with a hand-held camera where some of the high frequency content was smoothed out due to abrupt motions in the video.

As shown in Figure 5, both methods can resolve content at twice the frequency of a single LRF (each LRF contains frequencies up to π). As expected, the combined and display enhancement methods differ most for non-professional videos and are almost identical for computer generated videos, since they contain ideal flow fields and most high frequencies. Furthermore, the magnitude spectrum of our results follows more closely the target frames' spectrum in the horizontal modes than in the vertical modes. We attribute this difference to the fact that the SPSFs are more elongated in the vertical axis (Figure 2), and therefore do not span the vertical high-frequencies as well. Another factor may be that horizontal camera/object movements are more common in videos. These tests only show how much resolution enhancement can be achieved using our model, but do not take into account other factors such as flickering that might influence the perceived resolution. We thus complement this analysis with a user study (Section 5).

Display PSFs Previous approaches either use box functions for the display's PSFs [Templin et al. 2011; Didyk et al. 2010a] or model the display's specific PSFs [Berthouzoz and Fattal 2012]. Our method falls into the second approach. To evaluate the gain in modeling the display PSFs, we replace our acquired kernels by box functions and by circular gaussian kernels when computing the LRFs. We then compare the resulting displayed images (i.e., $\mathbf{W}L$, where now $\mathbf{W}$ contains the acquired display kernels). Figure 6 indicates that using the aquired kernels we can reconstruct up 14% more of the high frequency content. Thus, while modeling the display kernels requires computing different LRFs for each type of display, it also leads to an increase in the enhancement.

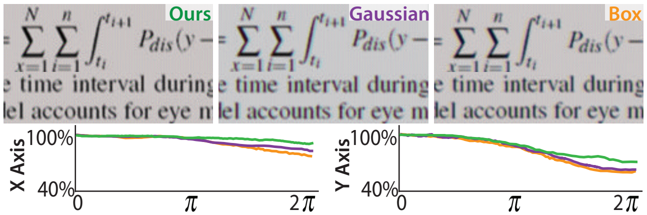


Figure 6: *(top) Example perceived frames for three scenarios. (bottom) Percentage of the magnitude spectrum that our kernels (green) achieve compared to gaussian (purple) and box (orange) kernels at every frequency for the horizontal and vertical direction.*

Comparisons. We include comparison videos to Templin et al [2011] in the supplemental material. Both approaches produce comparable resolution enhancement. However, Templin et al's method as described in [Templin et al. 2011] requires 90 frames per second input videos. Since our input videos are meant to be played at 30 frames per second, Templin et al's output videos play three times as fast as ours. Note that the goal of our method is not to surpass Templin et al's method in terms of resolution enhancement. Rather, our main focus is to achieve resolution enhancement for input videos that are *at* the display resolution by combining display resolution enhancement and super-resolution methods.

Computation Time. Our not optimized C/C++ code runs on an Intel i7 Core 2.8GHz. Our combined super-resolution and resolution enhancement method takes 6 seconds per input frame to generate LRFs of size 720 × 405. We observe a speedup by a factor 1.8 when using our combined approach versus each method separately.

5 User Study

We test the effectiveness of our resolution enhancement technique in a user study. We evaluate our approach on eleven 10 to 18 seconds long natural video clips showing scenes such as a man walking down a path or a rotating machine. The input videos are at the display resolution and we thus apply our combined approach to all of them. We generate our videos for $n=2$ and $n=3$ LRFs, such that we can display them at 60Hz and 120Hz respectively. We compare our videos to the input videos and to *sharpened videos*. We use an unsharp-mask to generate the sharpened frames which are given by $F + 0.08\ \nabla^2 F$ where F are the input frames. The sharpening filter increases the contrast of the edges, but it cannot add fine detail that increases the effective resolution of the video. We use the sharpened videos as a baseline technique to compare our method to. We

[1] http://www.vis.berkeley.edu/papers/vidEnh/

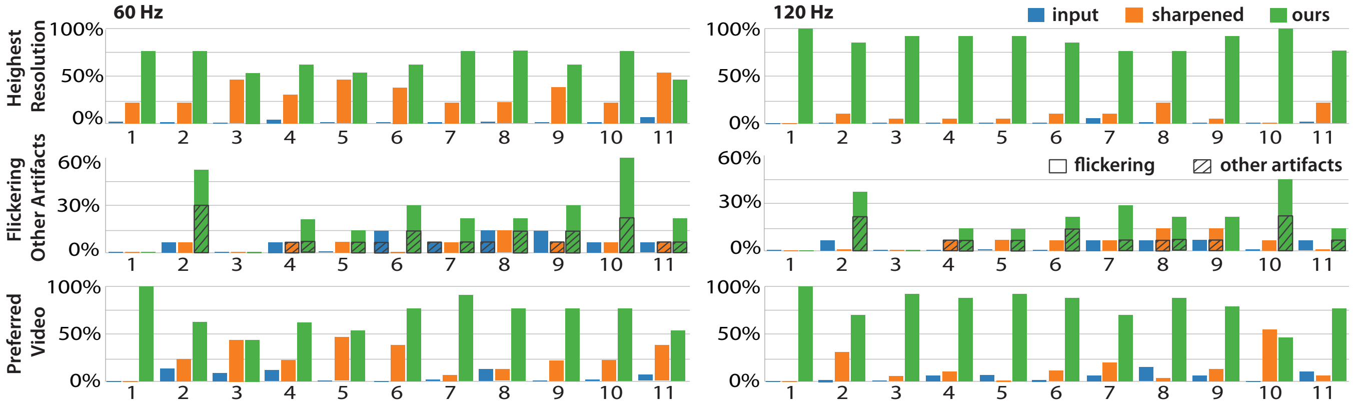


Figure 7: *The percentage of users selecting the input, the sharpened or our video as the video with highest resolution (top row) or as their preferred video (bottom row) for each of the 11 clips. (middle row) The percentage of users seeing flickering or other artifacts in the videos.*

did not compare to Templin et al.'s [2011] approach in this study, because their output videos would play three times as fast as our videos for 30 frames per second input videos.

26 participants took part in our study. They were naive regarding the goal of the experiment. In the first part of the study, subjects compared our videos to the input and sharpened videos and had to choose the video that contained most fine detail. The input, sharpened and our video were played simultaneously and were placed in random orders side-by-side. Users could replay them as many times as they wanted. For each viewer, half of the video clips were played at 60Hz and the remaining half at 120Hz. In the second part of the study, users viewed the same set of videos again. They were asked to mark all videos in which they perceived any flickering, aliasing or other artifacts and had to describe what artifacts they saw. Finally, after the users had to point out the artifacts in the video, we asked them to choose the video that they preferred between the input, ours and the sharpened video. There are three notable findings from our user-based evaluation:

1. Our videos perceived to have the highest resolution.
Figure 7 (top row) shows that users consistently perceived our videos to have higher resolution than the input or sharpened video. At 60Hz, for each clip, on average 71% (STD 11.6) of the users chose our video as having the highest resolution. As expected, this trend becomes stronger when the subjects viewed the videos at 120Hz. At 120Hz, for each clip, on average 88% (STD 8.7) of the users chose our video as the one with the highest resolution.

2. Some of our videos contain noticeable artifacts.
Figure 7 (middle row) shows that the subjects indeed saw some artifacts in our videos. Most complaints about artifacts were related to flickering at sharp edges, others were described as 'weird motion effects' and may be attributed to errors in the optical flow. Only two people mentioned aliasing artifacts for two videos. Aliasing can occur when the LRFs are not fused into a high-resolution frame. Our subjects mostly described the artifacts as minor and the fact that they also hallucinated artifacts in the input and sharpened videos, shows that they were scrutinizing the videos very carefully. Only for 2 out of the 11 clips, the subjects consistently perceived flickering/artifacts. To further remove flickering, we could use the post-processing step developed by Didyk et al. [2010a], but at the cost of losing some resolution enhancement.

3. Despite artifacts, users prefer our videos.
Even after subjects pointed out the artifacts in each video, they still consistently preferred our videos (Figure 7 (bottom row)). For each clip, on average 70% (STD 13.5) and 81% (STD 11.4) of the subjects preferred our videos when viewing them at 60Hz and 120Hz respectively.

6 Discussion

We presented a method that exploits properties of the eye for increasing the apparent resolution of videos. Our contributions and improvements over previous approaches [Templin et al. 2011; Didyk et al. 2010a] include the combination of display resolution enhancement with the super-resolution methodology, the derivation of the perceived video model that accounts for general shapes of the camera and display light elements, the treatment of degenerate cases that occur in the optical flow of natural videos and the analysis and treatment of non-uniform grid alignments.

Our approach strongly depends on the motion available in the video and the ability to track it. Our method therefore does not increase the perceived resolution of still or rapidly moving objects. Nevertheless, it does not introduce visual artifacts in such cases. Also, even under ideal settings the resolution enhancement is restricted to the direction of the motion. Despite these limitations, our study shows that users consistently perceive our videos as having higher resolution and prefer our videos to the input or sharpened videos.

Acknowledgments. We thank Shmuel Peleg for insightful discussions, and Krzysztof Templin for running the method described in Templin et al. [2011] on our input videos. This work was supported by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities, as well as the Microsoft New Faculty Fellowship Program.

References

ALLEN, W., AND ULICHNEY, R. 2005. Wobulation: Doubling the addressed resolution of projection displays. *SID 47*.

BEN-EZRA, M., ZOMET, A., AND NAYAR, S. 2004. Jitter camera: High resolution video from a low resolution detector. In *CVPR*, vol. 2, IEEE.

BERTHOUZOZ, F., AND FATTAL, R. 2012. Resolution Enhancement by Vibrating Displays. *ACM Trans. on Graphics (TOG) 31*, 2, 15:1–15:14.

BOCCIGNONE, G., MARCELLI, A., AND SOMMA, G. 2002. Analysis of dynamic scenes based on visual attention. *AIIA*.

BORMAN, S., AND STEVENSON, R. 2002. Super-resolution from image sequences-a review. In *Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on*, IEEE, 374–378.

BROX, T., BRUHN, A., PAPENBERG, N., AND WEICKERT, J. 2004. High accuracy optical flow estimation based on a theory for warping. *Computer Vision-ECCV 2004*, 25–36.

DAMERA-VENKATA, N., AND CHANG, N. 2007. Realizing super-resolution with superimposed projection. *CVPR 2007*, 1–8.

DIDYK, P., EISEMANN, E., RITSCHEL, T., MYSZKOWSKI, K., AND SEIDEL, H. 2010. Apparent display resolution enhancement for moving images. *SIGGRAPH 29*, 3.

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

ELAD, M., AND FEUER, A. 2002. Superresolution restoration of an image sequence: adaptive filtering approach. *Image Processing, IEEE Transactions on 8*, 3, 387–395.

HENDERSON, J. M. 2003. Human gaze control during real-world scene perception. *Trends in Cognitive Science 7*, 11.

IRANI, M., AND PELEG, S. 1990. Super resolution from image sequences. In *ICPR*, II: 115–120.

JAYNES, C., AND RAMAKRISHNAN, D. 2003. Super-resolution composition in multi-projector displays. In *Proc. of IEEE Int. Workshop on Projector-Camera Systems*.

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

PARK, S. C., PARK, M. K., AND KANG, M. G. 2003. Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine 20*, 3 (May), 21–36.

PATTI, A., SEZAN, M., AND TEKALP, M. 2002. Superresolution video reconst. with arbitrary sampling lattices and nonzero aperture time. *Image Processing 6*, 8, 1064–1076.

PLATT, J. 2002. Optimal filtering for patterned displays. *Signal Processing Letters, IEEE 7*, 7, 179–181.

TEMPLIN, K., DIDYK, P., RITSCHEL, T., EISEMANN, E., MYSZKOWSKI, K., AND SEIDEL, H. 2011. Apparent resolution enhancement for animations. *SCCG*.

YOULA, D. 1978. Generalized image restoration by the method of alternating orthogonal projections. *Circuits and Systems, IEEE Transactions on 25*, 9, 694–702.

Appendix: Non-Uniform Grid Arrangement Effect on Matrix Conditioning

The following analysis estimates the effect that the offsets between the LRFs have on the condition number of $\mathbf{W^TW}$ and $\mathbf{SW}$. It considers non-uniform grids composed of several (n) uniform grids with the same amount of grid points. Similarly to Section 3.2, the sub-grids are offsetted by js where $j = 1,2..$ (and as we discussed in the paper, s depends on the magnitude of the motion vector). We perform this analysis in one-dimensional space and for the case of $n = 3$. The same result applies for the case of $n = 2$.

In Fourier space equation $\mathbf{W}L = H$ becomes

$$\hat{S}_{dis}\left(\frac{\omega}{3}\right)\left(\hat{L}_1(\omega)+\hat{L}_2(\omega)e^{i\omega v}+\hat{L}_3(\omega)e^{i\omega 2v}\right)=\hat{H}\left(\frac{\omega}{3}\right). \quad (16)$$

The ω and $\omega/3$ result from the resolution difference between the LRFs and the high-resolution image H. The LRFs ($S_{dis,i}$) and the high-resolution frame contain discrete samples and therefore the functions above are 2π-periodic. Thus, by evaluating it on $\omega, \omega+2\pi$ and $\omega+4\pi$, we get the following three constraints at ω

$$\begin{bmatrix} 1 & e^{i\omega s} & e^{i\omega 2s} \\ 1 & e^{i\omega s}e^{i2\pi s} & e^{i\omega 2s}e^{i4\pi s} \\ 1 & e^{i\omega s}e^{i4\pi s} & e^{i\omega 2s}e^{i8\pi s} \end{bmatrix}\begin{bmatrix} \hat{L}_1(\omega) \\ \hat{L}_2(\omega) \\ \hat{L}_3(\omega) \end{bmatrix}=\begin{bmatrix} \hat{H}_S(\frac{\omega}{3}) \\ \hat{H}_S(\frac{\omega+2\pi}{3}) \\ \hat{H}_S(\frac{\omega+4\pi}{3}) \end{bmatrix}, \quad (17)$$

where $\hat{H}_S = \hat{H}(\omega)/\hat{S}_{dis}(\omega)$. As we mentioned earlier, we do not elaborate here on the implication of inverting the SPSFs and refer the reader to Berthouzoz and Fattal [2012].

In order to investigate the condition number of the matrix in (17), we consider the following normalized column vectors

$$\mathbf{v}_s^j = [1, e^{i2(j-1)\pi s}, e^{i2(j-1)\pi s}]^T/\sqrt{3},$$

where $j = 1,2,3$ and define a 3-by-3 matrix, denoted by $\mathbf{A}_s$, as the one containing these vectors as its columns. The matrix $\mathbf{A}_s$ differs from the one in (17) by an overall factor of $1/\sqrt{3}$ and the scalars, $e^{i\omega s}$ and $e^{i\omega 2s}$, multiplying the second and third columns of (17). The condition number of $\mathbf{A}_s$ can be defined as follows

$$\max\|\mathbf{A}_s\mathbf{u}\|/\min\|\mathbf{A}_s\mathbf{w}\| \text{ s.t. } \|\mathbf{u}\|,\|\mathbf{w}\| = 1.$$

Therefore, an overall multiplication by a scalar does not change this ratio. Furthermore, a multiplication of the column vectors by unit scalars does not change the magnitudes $\|\mathbf{A}_s\mathbf{u}\|$ and $\|\mathbf{A}_s\mathbf{w}\|$. Therefore, we can use the matrix $\mathbf{A}_s$ to model the one in (17) in terms of their condition number, because both matrices share the same condition number.

It is easy to see that $s = n^{-1}$ (plus any integer) makes $A_{n^{-1}}$ the DFT matrix operating in R^3 which is a regular matrix with condition number of one - an ideal setting for solving any linear system. However, as s becomes close to an integer, $\mathbf{v}_s^2$ and $\mathbf{v}_s^3$ converge to the first column and therefore the matrix severely degenerates. Specifically, as $s \to 0$ (or any other integer) both columns $\mathbf{v}_s^2$ and $\mathbf{v}_s^3$ equal $\mathbf{v}_*^1 + O(s)^2$, which is dictated by the rate the terms $i\sin(2(j-1)\pi s)$ converge to zero.

We estimate the condition number of $\mathbf{A}_s$, when s approaches an integer, by evaluating $\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^1\|/\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^2\|$ (or $\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^1\|/\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^3\|$) which gives us a lower bound (we saw above that the condition number is given by the maximal value of this ratio).

The nominator in this case becomes $\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^1\| = 3+O(s)$ since $\mathbf{v}_{n^{-1}}^1$ is all $1/\sqrt{3}$ and, as s is close to an integer, the components of the matrix $\mathbf{A}_s$ are all $1/\sqrt{3}+O(s)$. On the other hand, since $\mathbf{v}_{n^{-1}}^j$ with $j = 1,2,3$ form an orthonormal basis, Parseval's identity applies

$$1 = \|\mathbf{v}_s^2\|^2 = \sum_{j=1}^{3}|\langle\mathbf{v}_s^2,\mathbf{v}_{n^{-1}}^j\rangle|^2 = 1+O(s^2)+|\langle\mathbf{v}_s^2,\mathbf{v}_{n^{-1}}^2\rangle|^2+|\langle\mathbf{v}_s^2,\mathbf{v}_{n^{-1}}^3\rangle|^2,$$

which results from $\mathbf{v}_s^2 = \mathbf{v}_{n^{-1}}^1 + O(s)$. This gives us that $|\langle\mathbf{v}_s^2,\mathbf{v}_{n^{-1}}^2\rangle|^2$ and $|\langle\mathbf{v}_s^2,\mathbf{v}_{n^{-1}}^3\rangle|^2$ are both $O(s)$. The same decomposition to $\|\mathbf{v}_s^3\|^2$ gives us that $|\langle\mathbf{v}_s^3,\mathbf{v}_{n^{-1}}^2\rangle|^2$ and $|\langle\mathbf{v}_s^3,\mathbf{v}_{n^{-1}}^3\rangle|^2$ are also $O(s)$. We get

$$\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^2\|^2 = \sum_{j=1}^{3}|\langle\mathbf{v}_{n^{-1}}^2,\mathbf{v}_s^j\rangle|^2 = 0+2O(s),$$

where the first term vanishes since $\mathbf{v}_{n^{-1}}^2$ and $\mathbf{v}_s^1$ are orthogonal. The same steps applied for $\mathbf{v}_s^3$ give $\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^3\|^2 = O(s)$.

We finally conclude that the condition number of $\mathbf{A}_s$ is greater than

$$\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^1\|/\|\mathbf{A}_s\mathbf{v}_{n^{-1}}^2\| \approx (1+O(s))/(O(s)) = O(s^{-1}).$$

[2] The star sign $*$ indicates that this relation holds for every value.