Device Dependent Image Construction

for Computer Graphics

by

Robert Victor Klassen

A thesis
presented to the University of Waterloo
in fulfillment of the thesis
requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada.
July, 1989

Borrower's Page

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign your name in the space below, giving your address and the date.

Abstract

All computer-generated images are displayed by intensifying pixels on a display. Each pixel, which extends spatially and temporally, has two relevant properties: its position and its intensity profile. The interaction between position and image quality, which can produce the aliasing of high frequencies onto low ones, has been extensively explored and very successful treatments for the resulting artifacts are well known. There is little practical understanding of the interaction between intensity profile and image quality.

This thesis examines the interaction between pixel intensity profiles and image quality, taking into account the spatiotemporal characteristics of the human visual system. The interaction is examined by considering the general problem: given a device with given pixel locations and intensity profiles, what is the set of intensity values that best represents a given spatiotemporal image. Within the restricted but practical case in which pixel locations are periodic in space and time and pixel intensity profiles are identical, two solution techniques are explored. One directly minimizes discrepancies between the desired image and an image generated by a display device. The other chooses pixel intensities to minimize differences in the Fourier domain, with the differences weighted by the corresponding sensitivities of the human visual system.

These two techniques are explored in detail for pixels with an exponential temporal intensity profile and a Gaussian spatial profile. Each method is examined in several different norms with pixel intensities constrained and unconstrained. (Constraints are relevant because the contrast possible using unconstrained fitting is very restricted for some devices.) These results are calculated assuming temporal degrees of freedom independent of the spatial ones. Under the same assumption, the spatial intensity profiles were examined. Throughout these calculations, algorithms that manipulate circulant matrices provide a computationally effective means for determining pixel intensities. When spatial and temporal degrees of freedom are taken together these algorithms can no longer be used because the spatial and temporal responses of the human visual system are not separable. Since solutions require use of less efficient numerical methods, the emphasis in that part of the thesis is on differences between the unseparated solutions and those that are produced using separable approximations.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada through a post-graduate scholarship, infrastructure and strategic grants; by the Province of Ontario government through an Ontario Graduate Scholarship and a BILD grant to the University of Waterloo Computer Graphics Laboratory; by the University of Waterloo Institute for Computer Research through a scholarship and grants to the laboratory; by the Ontario Information Technology Research Centre through a scholarship and grants to the laboratory; and Digital Equipment Canada through equipment donations.

The committee that approved this thesis consisted of Richard H. Bartels, Kellogg S. Booth (co-supervisor), Andrew R. Conn, William B. Cowan (co-supervisor), Franklin C. Crow (external examiner) and Wei-Pai Tang.

Richard Bartels pointed out the separability allowing a two-dimensional least squares problem to be carried out as two one dimensional problems in a tensor product situation. William Cowan deserves the blame for thinking this topic was something worth doing, and pushing me until it had evolved to the current form. The frequency space minimization is at least partly his doing. Lyn Bartram was the second observer for the figures in Chapter 7. Thanks also to Lyn for seeing the original copy of this thesis through the copying and binding process.

To my wife Sue and son Nathan, I owe many months of my time. Their patience through this work, particularly the writing, has been most appreciated. Thanks are also due Sue for help with final modifications and proof-reading. I take responsibility for anything she missed.

Dedication

To my wife, Sue. Now and forever.

Contents

List of Symbols

ξ	spatial or temporal independent variable
ς	spatial or temporal frequency
t	time
x, y	position
ω	temporal frequency
k, k_x, k_y	wave number (spatial frequency)
n, m	index in Fourier series
ϕ	phase
a_n, b_n	real Fourier series coefficients
A_n, ϕ_n	Fourier series coefficients as amplitude and phase
τ	temporal phase shift
κ	spatial phase shift
i	$\sqrt{-1}$
$\mathcal{S}[g(\xi), \varsigma_n]$	the nth complex Fourier series coefficient of g.
$\mathcal{F}[g(\xi), \varsigma]$	Fourier transform of $g(\xi)$, using ς as conjugate of ξ
$\mathcal{F}_c, \mathcal{F}_s$	sine and cosine transforms
n, p, q	index in infinite series
i, j, l	index in finite series; matrix index
w_k	weight applied to kth frequency
$f \circ g$	convolution of f with g
λ	exponential decay constant
σ	Gaussian spread coefficient
$\delta(t)$	delta function
$\mathbf{M}$	matrix with elements m_{ij}
$\mathbf{v}$	vector with elements v_j
z^*	complex conjugate of z
$\mathbf{M}^{\mathbf{T}}$	transpose of $\mathbf{M}$
$\mathbf{M}^{\mathbf{H}}$	Hermitian transpose of $\mathbf{M}$
$\mathcal{B}(x, t)$	infinite sum of basis functions (pixel shape) all of which have the same position modulo the periodicity of the image.
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of $\mathbf{A}$ and $\mathbf{B}$

Introduction

A major goal of computer graphics in general, and of this thesis in particular, is the improvement of the quality of the images produced. There are two main components in today's computer image generation systems: modelling and rendering. The modelling system typically produces a data structure which describes the scene to be portrayed, in terms of objects and their properties such as position, velocity, orientation, texture, reflectance, shape and size. The rendering subsystem uses the results of modelling to produce images. The nature of the modelling subsystem usually determines how difficult it is for the user to specify the scene, while the nature of the rendering subsystem controls the quality of the resulting image.

The specification of a scene produced by a modeller is usually sufficient for finding the intensity everywhere in a continuous region of space and time. Because of the discrete nature of display devices, the images produced as a result of rendering only approximate the continuous specification. Display devices consist of pixels, which are specified by locations and intensities. The intensity distribution associated with the pixel determines how the intensity specified affects points other than the centre of the pixel. For example, on a CRT, the intensity distribution of a pixel can be approximated by a Gaussian distribution in space, and an exponential decay over time. The value specified at one pixel affects not only the intensity at the point of greatest influence of the pixel (in space), but also the intensity of a region surrounding the pixel, extending past the centres of several neighbouring pixels. The value specified continues to have an effect on the intensity within the region of influence of a pixel for a period of time which varies from a few microseconds to several seconds. It is typical in computer graphics to specify the value of the image at the point corresponding to the centre of a pixel, ignoring the fact that the values specified for neighbouring pixels affect the intensity there.

An important part of rendering is scan conversion, in which a discrete representation of the continuous image is found. Ideally, scan conversion produces a representation that is visually as close as possible to the continuous original. Two kinds of artifacts make this difficult. One is aliasing of high frequencies to low, and

is affected by the locations of the pixels. This is discussed in Chapter 2. The other is pixel structure, which is affected by the intensity distributions of the pixels. This thesis explores methods of improving the quality of the results of the scan conversion process, primarily in reducing or eliminating pixel structure artifacts. Depending on the specification of the original image, these methods sometimes reduce aliasing artifacts at the same time. Improved quality implies increased similarity between the image desired and the image displayed. This can be measured in various ways. In this thesis visual quality is emphasised; a simplified model of the human visual system is used to approximate the visual system's response to errors in the image, which is then minimized. This model uses the frequency response of the visual system, and is used to assign different degrees of importance to errors in the image according to their frequency.

The process of minimizing the error is described in terms of solving a matrix vector problem in which the matrix is circulant, or nearly so. Efficient techniques which take advantage of the circulancy are used to find the solution.

In order to compensate for pixel intensity distributions, it is typically necessary to use intensities which are outside the range of intensities in the continuous input image. Taking as an example the exponential decay of a CRT, the intensity remaining from previous frames is subtracted from the intensity required. In all but the first frame, the pixel value needed at a frame is less than the image intensity. If the intensity required in a particular frame is less than that remaining from previous frames, then the pixel value needed is negative. Since negative intensities of light don't exist, the pixel value needed is outside the range of intensities which the device can produce. To keep the device intensities within the range of intensities which the device can produce it is necessary either to constrain the device intensities in the minimization or to place restrictions on the local variations in intensity in the input image. (The intensity can never drop in one frame to less than what remains from the previous frame.) A method of deriving these restrictions is given in general, and the derivation is carried out completely for the case of a device with intensities falling off exponentially with time.

Two recurring themes in work on rendering algorithms have been efficiency and quality. Running time seldom drops as a result of increased efficiency, or faster hardware; the extra time available is usually consumed improving the quality. Such improvements are generally quantitative, such as increased resolution or higher scene complexity. Qualitative improvements resulting from new techniques tend to increase the computation time for a given resolution of scene complexity, but they sometimes reduce the resolution or scene complexity required to give the same level of quality in appearance. Qualitative improvements are often followed by a search for more efficient means of achieving the same end, so that the total computation time required

for a given scene complexity and resolution can be brought back to what it was before the improvement was made in quality. The techniques presented in this thesis, while not as efficient as possible, examine the limits to quality. Their purpose is to discover what is important for the very best quality. Efficiency will come later. Thus, they are intended to complement, not replace, the better rendering techniques already in use.

The reader should by this point have an idea of what to expect in this thesis. The remaining part of this chapter contains a more complete outline.

In computer graphics, rendering is concerned with making images from scene descriptions. In this thesis the term image is extended to mean any function mapping positions in time and/or space to intensities. Thus animated images are included. The image may have any of a number of representations, but it must eventually be displayed on a real device. The class of device with which we are concerned is termed *raster* devices. On such a device, the values associated with individual pixels are specified, and the intensity throughout the device is given by linear combinations of pixel intensity profiles. ("Pixel" is also extended to include the temporal dimension). Different devices have a variety of different intensity profiles. The pixel may have a uniform influence over a localized region, with a sharply defined edge. The liquid crystal display is a good example. Or the pixel may have a strong influence at the centre of a region, with the effect diminishing in all directions, or in one direction preferentially. For animated images, the functional dependence must be specified in three dimensions, one temporal and two spatial. For example, on a good quality black and white CRT, the intensity distribution is well approximated by a Gaussian distribution vertically and an exponential decay over time. Horizontally, the intensity profile is less well approximated by a Gaussian, particularly on poorer quality CRTs. On a colour CRT, the horizontal and vertical intensity profiles are complicated by the shadow mask, while the decay times of the different coloured phosphors in general differ.

Two features of raster devices affect the quality of the resulting image: the locations of the pixels, and the intensity profiles of the pixels. The positioning of the pixel centres can lead to aliasing of high frequency information to lower frequencies: this effect is well known, and effective solutions have been found for it. The effects of the intensity profiles of the pixels have received very little attention.

This thesis considers the effects of pixel shape, ways of compensating for undesirable effects, and ways of simulating desired pixel shapes. Two pixel shapes are considered as examples: the Gaussian pixel typical of the spatial profile of a pixel on a CRT, and the exponential decay function typical of the temporal profile of a CRT pixel. Both of these involve a compromise between blurring the image (leaving trails in the case of the temporal profile) and introducing unwanted high frequency

information. Over-focussed monitors fail to produce a flat field when a constant value is set in all of the pixels, since the gaps between scanlines are visible; images displayed on underfocussed monitors fail to appear as sharp as they could. Monitors with a short persistence phosphor have an annoying flicker, while monitors with long persistence phosphors can have pronounced after-images. A correctly focussed monitor with a moderate persistence phosphor does not have the problems of over-focus or flicker, but it does still show some after-images, and the image is less sharp than it might be, unless steps are taken to compensate for these effects. Simulated pixels are both a tool for studying the effects of pixel shapes and the quality of a method of compensating for them, as well as a paradigm for image enlargement.

Controlling the effects of pixel shape goes beyond choosing beam parameters for CRTs. First, there are devices, such as liquid crystal displays and thermal transfer printers, that have pixels with significantly different pixel shapes. Secondly, as device pixels become more and more densely packed spatially (and perhaps temporally), it may become possible to simulate logical pixels of arbitrary shape at reasonable resolution using multiple physical pixels. Thirdly, because the cost of computing images to full device resolution grows as the square of the device resolution, it will sometimes be more cost effective to compute the image to reduced resolution and then enlarge it using simulated pixels. Finally, an image computed for a device with one pixel profile does not look its best when displayed on another, unless appropriate compensation is made.

Image enlargement has usually been viewed as a variant of interpolation. This can introduce overshoot and undershoot when the interpolation attempts to follow apparent discontinuities in the image. More importantly, it leaves entirely implicit assumptions about the intensity profiles of the pixels in the original image. If the low resolution raster is a discretization of a piece-wise cubic function then cubic spline interpolation may be the appropriate choice. More typically the image is derived from a discontinuous function which has been filtered to remove aliasing artifacts. If the image has been tailored to look its best on one particular device, then the pixel shape of this device should be taken into account in the enlargement. Commonly, images are enlarged by pixel replication, producing large, square composite pixels. Enlargement by using simulated large pixels rather than some model of interpolation gives a more faithful imitation of the effect of moving closer to the original image.

Brute force methods of compensating for pixel shape are simply described and require no advanced mathematics. The most naive approach is as follows: as the intensity value of any pixel is computed, the residual effect on its neighbours is calculated, and an appropriate amount is subtracted from the values of those pixels. The amount of compensation can be calculated to match the values of the ideal and device images at the pixel centres, to match the average intensities between pixel

centres, or to match some interpolation of the ideal and device images between pixel centres. Because a pixel value is changed after it is used to compute corrections for the neighbours of that pixel, second order corrections must be computed to correct for the effect of the first order corrections. This iterates until the error is smaller than some tolerance, or it ceases to diminish from iteration to iteration. Convergence depends on the pixel profile.

The approach described above is known as Jacobi iteration, a method used in the solution of large matrix-vector problems. To improve on it requires more mathematical sophistication. With the problem cast as a matrix-vector problem, more direct methods may be used, especially since the problem size need not be great enough to justify iterative techniques. In such a system, the matrix depends on the device, the right-hand side vector depends on the desired image, and the solution vector gives the intensity values at the pixels. If we assume that the image is periodic (with a sufficiently large period that this imposes no constraints on the actual image), the matrix is circulant, in which case fast inversion techniques, involving the use of the Fast Fourier Transform or FFT, are known. If the matrix is not square, then the problem is one of minimizing the error, and the choice of norm for measuring the error is important both computationally and in the results obtained.

The human visual system perceives different spatial and temporal frequencies with different proficiency. It makes sense to take this into account when minimizing errors in an approximate solution. In doing so, the Fourier coefficients of the solution are compared with those of the ideal image, and the differences are weighted by the sensitivities of the human visual system at corresponding frequencies. Various approximations to the sensitivity function may be used. Those approximations which involve the product of independent functions for spatial sensitivity and temporal sensitivity lead once again to circulant systems. Those which do not are considerably more expensive to compute, and produce surprisingly similar results.

Regardless of whether the minimization is performed in the Fourier domain or directly, constraints on the solution may be required to avoid values (such as negative intensities) which cannot be reproduced on the device. For Jacobi iteration this involves using either the correction term predicted or the one which causes the values to reach their limits, whichever is smaller at a given iteration. (Other, more complex, methods for constraining iterative solutions also exist). The alternative to constraints is a reduction of contrast in the input. Depending on the circumstances, one or the other may be preferred.

The main purpose of this thesis is to discuss methods of finding the set of intensity values that are visually best for a given image and a raster device with given pixel intensity profile. The presentation begins with non-mathematical background

material. This consists of Chapters 2 and 3, describing elimination of aliasing artifacts and properties of the human visual system, respectively.

There are a number of assumptions, particularly about the nature of the display device, that are required to make the problem tractable. These are outlined in the Chapter 4, which defines the problem, its scope and limits.

To solve the problem as defined, a number of mathematical tools are useful. One of these is Fourier analysis, which is well known. Because it is used throughout the thesis, a number of standard results appear in the appendix for reference. The other major tool is matrix algebra, particularly with circulant matrices. To establish notation, a number of definitions from matrix algebra are given. These are followed by definitions and results regarding the less familiar circulant matrices and a new form of matrix, termed "pseudo-circulant". Several minimization algorithms leading to circulant matrices complete the mathematical toolkit in Chapter 5.

Even with restrictive assumptions about the device, there remain a number of alternative interpretations, which are explored in Chapter 6. The solution specifies the intensity only at a finite number of places, so that a perfect match between the image desired and the one produced can only be guaranteed at this many positions (not necessarily the same positions as those at which the intensity is specified). What then do we want to match? Should the image match perfectly at the positions at which we specify the intensity? Perhaps the total intensity over the region between such positions should agree. Alternatively, there are various interpolations possible, which can be well matched throughout. Of these, trigonometric interpolation should be closest to the original image, if the image is not prone to aliasing. For any of these possibilities, the solution may involve unattainable intensity values. But with some care, it is possible to place restrictions on the contrast of the input so that this does not happen. At this point, the problem remains abstract, with no pixel profiles specified; examples of specific profiles follow later.

In order to know whether the methods work, and to make a visual check on the algorithms, it is important to be able to display the results. Usually they are displayed at a relatively low resolution and viewed from a distance. Pixels of intensity profiles other than those provided by the device can then be accurately simulated for comparison purposes. It is also possible to adjust the Gaussian spread of the phosphor dot without changing the monitor. (The spread can be changed by adjustment of the focus control – but due to the idiosyncratic behaviour of hardware the software approach is preferred). In the temporal domain, the phosphor decay is not adjustable after the monitor is manufactured. With a sufficiently short persistence phosphor, and a monitor capable of very high refresh rates, it is possible to simulate any decay rate (and in fact any other shape of temporal profile) well enough that any errors in

the simulation are invisible. Few frame buffers are capable of driving a monitor at sufficiently high rates for such simulations, but one which is capable is the Ikonas RDS/3000. Its use in these simulations is described in the seventh chapter.

One dimensional examples are sufficient to allow us to explore the differences of different minimization techniques, and the results of changes in the pixel spread. Two examples are used in Chapter 8 to illustrate the techniques developed earlier: they are the Gaussian, and the exponential decay.

Chapter 9 extends the techniques from one dimension to two. All of the one dimensional techniques can be extended to two spatial dimensions using two passes, one for each dimension. Most of them can be extended in this way to include a temporal dimension as well. The exception is the method which uses the sensitivity of the human visual system to weight terms in the Fourier series according to their relative visibility. This exception occurs because the visibility of information at a given temporal frequency depends on the spatial frequency and conversely. As a result of the inter-dependence, a two pass method is not available, and what is more, the efficiency which results from circulancy in the one dimensional case, and carries through to the two-pass method, is not available because the circulancy is lost. It is a fortunate result that the weighting makes sufficiently little difference to the final answer that this method is not, in the end, the method of choice.

In summary, the thesis consists of three parts. First, there is the background material in Chapters 2 and 3, consisting of prior work in computer graphics and vision research. Second, a theoretical formalism for finding the best raster representation of an image is presented. This consists of a formal definition of the problem, mathematical tools and algorithms, and a set of approaches available for solving the problem (Chapters 4 through 6). Finally, Chapters 7 through 9 contain practice with and experience gained from working with the formalism, beginning with methods of simulating unavailable devices, followed by concrete examples of finding solutions in one and two dimensions for devices with Gaussian and exponential pixels.

Aliasing Artifacts

Causes, Effects, Prevention

The descriptions of scenes which are provided to rendering systems are normally sufficient to describe an image everywhere within a continuous region. Devices used to display these images consist of pixels, typically arrayed on a regular grid. When the resolution of the grid is insufficiently high to represent the highest frequencies in the image, there is the potential for aliasing of the high frequencies to low frequencies in the image as it is displayed.

Artifacts caused by aliasing of high frequencies as lower ones are well known in computer graphics. Very effective means of preventing them are also known. If an image contains aliasing artifacts, then any problems related to the pixel intensity profile are masked. Because we are interested in high quality images, it only makes sense that the images be free of serious aliasing artifacts, although it is sometimes worthwhile to trade aliasing for improvements elsewhere. In this chapter, the sources of aliasing, types of artifacts produced, and methods of reducing and preventing aliasing are discussed.

Rendering for raster displays involves a process known as scan conversion. Scan conversion is the process by which an image defined on a continuous domain is converted to a discrete representation and stored as an array of intensity values. Scan conversion often consists of specialized algorithms designed to determine quickly which pixels (array elements) are influenced by each of the primitives (elementary geometric objects) that make up the image. To save storage, the only record many such algorithms keep of a primitive that has already been processed is the intensity values of pixels that were affected. When more than one primitive affects a pixel, it is not possible using such algorithms to determine what contribution the first primitive should give when the second is being processed. Consider, for example, the depth-buffer algorithm, which stores at a pixel the colour of the nearest primitive covering the centre of the pixel, and the depth of that primitive. This algorithm fails to find

the full information at certain pixels, particularly along the edges of primitives which occlude other primitives. This is illustrated by two scenes: the first consists of a single rectangle covering the entire visible region, behind a stack of triangles which are lined up in such a way that each one exactly hides the one beneath it, the second scene is like the first, except that the triangles are oriented randomly. For the first scene, pixels along the edges of the frontmost triangle should have their colours determined from the colours of the rectangle and the frontmost triangle, while for the second, their colours should be determined from the colours of the frontmost and whichever triangle is visible beyond the edge of the frontmost. Because the triangles are processed in an arbitrary order, the second nearest triangle at a pixel is not known when that pixel is drawn.

Better algorithms generally determine which primitives influence each of the pixels that make up the array, or raster, without wasting too much time considering primitives far from the pixels in question. In this way, full information about a given pixel may be known when the value for that pixel is computed.

If one part of scan conversion is determining which primitives affect a pixel, the other is finding an intensity value for the pixel. The simplest way of choosing which primitives affect a pixel is sampling. For each point in the array, the primitive at the corresponding position is computed, and its colour at that position is used as the raster pixel value. Such algorithms as the depth buffer algorithm are forced to use sampling. Since a pixel is associated with a point rather than a region, only the nearest primitive to the eyepoint is relevant. and only primitives which cover the centre of a pixel need be considered. It was early recognized that straightforward sampling can lead to artifacts. Sampling artifacts appear in a variety of guises. As a test case, Crow used a mathematically defined pattern (nested parabolae, gradually shallower and closer together), which produces very obvious Moiré patterns in parts of the image where the parabolae are closer together than two pixels. Other artifacts include staircasing of lines, edges and highlights, broken lines, and disappearing objects and highlights. In moving pictures, the staircases crawl along the lines, and objects which are missed in one frame may reappear in another, apparently blinking on and off [Crow1976]. Small objects can also change their size and apparent shape as they move.

Catmull has been credited with making the first attempt at preventing such artifacts [Catmull1974]. Catmull mapped a texture onto a surface and found that where the line of sight came near to tangential to the surface, the texture varied rapidly within the area corresponding to one pixel. This led to patterns appearing on the surface which did not appear in the texture. Catmull noted that the cause of the problem was high frequency noise in the texture, higher than the frequencies representable by the device. He found that averaging the intensity of the texture over the region in texture space corresponding to a pixel's region of greatest influence

attenuated the high frequency information sufficiently to make the images acceptable. In this way, what is known as box filtering or area averaging was introduced to computer graphics. In certain cases, box filtering can be implemented exactly (that is, to within machine precision), but in other cases, it has been implemented using either geometric approximations or approximations based on supersampling. Box filtering is easily understood, and can be implemented quite efficiently with a minimum of clever tricks. It also works very much better than not filtering at all.

Crow pointed out that sampling artifacts are the result of aliasing, in which frequencies at or above half the sampling frequency appear aliased as lower frequencies [Crow1976]. To demonstrate the weakness of box filtering, he used images with very thin, bright highlights, slightly tilted from vertical. These images were typical of the better quality images being produced at the time, and exhibited aliasing in the form of dotted highlights, even when box filtered. The solution he proposed was to remove enough of the high frequency information from the image to make the aliasing artifacts invisible. Filters predicted to be ideal from the body of knowledge about signal processing may be better than what is needed visually, and computationally too expensive, or may cause an image originally containing only positive intensities to have negative ones. Crow compared several simple filters and evaluated the results visually [Crow1981]. By doing this experiment he was able to find a filter which produced an acceptable image on his CRT. It is not clear how much the use of a CRT affected the choice of visually best filter. The methods in this thesis could be used to find device dependent compensation to filters found using Crow's methods.

A simple illustration can make the problems of undersampling clear. Consider an image whose intended intensity is constant in y and a sinusoidal function of x. If this is sampled at a frequency of greater than two samples per wavelength, the resulting image appears to have the same frequency as the original. The intensity profile between pixels depends on the intensity profile of the pixels but the dominant frequency of the waveform is the same. When the sampling frequency is less than two samples per wavelength, the image appears to contain lower frequency information. Figure 1 shows the one dimensional intensity profile of such an image, with the sampling rate smoothly dropping from left to right. At the left the image is sampled at a sufficiently high frequency, while at the right the sampling frequency is too low to represent the image. A theorem due to Whittaker, and later attributed to Shannon and Nyquist, states that for any function sampled with inter-sample spacing σ, there is a unique interpolating function which has no singularities, and has in its Fourier series representation no terms with period less than 2σ [Whittaker1915]. From this we may conclude that there is always a function with no frequencies higher than $1/2\sigma$ which fits any set of evenly spaced points, spaced apart a distance σ.

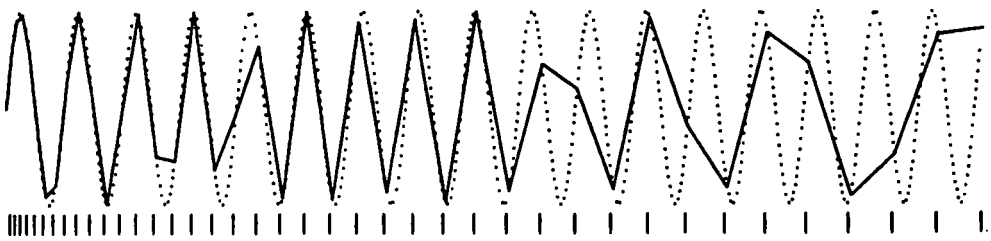

Figure 1: A single sinusoid (dotted curve) sampled at a rate which decreases linearly from left to right. Tick marks indicate the locations of sample points. Solid lines connect the sample points. At the left any interpolation of the sample points may contain no lower frequencies than the sinusoid being sampled, and the sampling clearly appears to be primarily of the same frequency as the sinusoid. When the sampling rate drops below the twice the frequency of the sinusoid, an alias appears, in the form of a variation at half the frequency of the sinusoid.

Aliasing results when the original function contains frequencies greater than $1/2\sigma$: the interpolation of the sampling need not contain frequencies greater than $1/2\sigma$, and the visual system does not choose to use an interpolation which introduces them. For functions specified over a finite domain of length Δ, with N samples over the domain, the highest frequency which may be represented without aliasing is $N/2\Delta$. For historical reasons this is known as the Nyquist rate.

If high frequency information is the problem, then the solution, known for some time in the field of signal processing, is to remove the high frequency information by filtering. The most common approach in computer graphics is linear filtering, defined as the convolution of the image with a filter function, yielding a new function with frequency characteristics that depend on the filter, and those of the original image.

$$I'(x,y,t) = \int I(x',y',t') f(x-x',y-y',t-t')dx'dy'dt'$$

In theory, the integral is over all space and time. In practice, the image is only non-zero over a finite region, and the filter is usually non-zero over a smaller region. Area averaging (such as was done by Catmull) is equivalent to filtering with a filter function which is a pixel-sized square in two dimensions, or cube in three (hence the name box-filtering). The attenuation of high frequencies by the box filter is proportional to the frequency and to the filter width. Better filters have less attenuation at low frequencies and more at higher frequencies. The best filter from the point of view of frequency response is the sinc function $\sin(ax)/ax$, which completely attenuates all frequencies below its cutoff frequency, a. Unfortunately it is impractical, since it has negative lobes (which may result in negative intensity

values near discontinuities), and does not die off very quickly. Other popular filters include conical and pyramidal filters, and various approximations to Gaussians or sinc functions which are non-zero over a relatively small region.

Antialiasing algorithms designed to use filtering as a means of reducing high frequency information generally consist of two parts. The first part decides which primitives affect a pixel, or which pixels are affected by a single primitive. The second part decides how to assign intensity values given the information found in the first part and the type of filter being used.

Deciding which primitives affect a pixel involves solving the hidden surface problem to find all of the primitives visible anywhere in the neighbourhood of a pixel, and in the better algorithms, finding the parts of the neighbourhood in which each such primitive is visible. Often the information found by the first part is only approximate. The set of primitives might be incomplete, and the areas in which each is visible are often only known approximately.

Assigning intensity values given a solution to the hidden surface problem involves finding the intensities associated with the visible primitives, and combining them in some way to provide the intensity of the filtered image. One way of combining them is to weight their average by the area they cover: this is box filtering. Other methods generally weight the intensities of regions in the image close to the points corresponding to the pixel centres more heavily. A large number of algorithms use one form or another of supersampling, in which the region affected by a pixel is sampled at a high resolution, and then the weighted or unweighted average of the samples is used as the value for the pixel. This is a simple form of numerical integration of the convolution integral. Any rendering algorithm can be used to do supersampling: the algorithm can compute as if for a higher resolution device, and then the pixel values in the final raster can be computed from weighted or unweighted averages of the pixels values in the corresponding neighbourhood in the high resolution raster. A problem with supersampling on a regular grid is that high frequencies contained in the image may be aliased in the supersampled version, and averaging does not attenuate the aliases any more than it does the legitimate low frequencies.

Most of the algorithms which do not use supersampling are limited to a restricted set of primitives for which the geometric properties of the primitives make it possible to compute the convolution integral directly. In many of these algorithms the filter is a box filter, which reduces the convolution integral calculation to finding the intersection area of the primitive with a square region corresponding to one pixel on the screen. Two examples of such algorithms are the line algorithm of Barros and Fuchs [Barros1979] which uses box filtering, and the polygon algorithm of Feibush,

Levoy and Cook [Feibush1980], which permits arbitrary filters.

Until 1984, the implicit assumption in antialiasing was that the way to deal with the problem was to remove high frequency information from the signal. At that point Cook, Porter and Carpenter introduced the idea of distributed ray tracing, and with it stochastic sampling [Cook1984]. Yellott had observed that the eye does not have a significant amount of aliasing in the image after its sampling, in spite of the low sampling density in regions away from the fovea (the region of high cone density corresponding to the centre of view) [Yellott1982]. He found that the Poisson disk distribution of the cone positions caused a loss of coherence in aliases. The example he used was a regular grating of bars, i.e., a square wave in one dimension. When this is sampled on a regular grid at a frequency less than twice the fundamental frequency of the grating, an alias appears: information at a well defined frequency below the cutoff gives the appearance of bars at a different frequency. When the same image is sampled with a Poisson disk distribution, the alias is converted into broad-band noise. As a result, the aliasing is lost amid other broad-band noise.

Stark had earlier found that the much more quickly computed jittered distribution contains a similar frequency distribution, and therefore produces similar results [Stark1977]. Cook reasoned that if incoherent sampling could eliminate aliasing in the eye, then it should work for displays as well. Using Stark's jittered distribution, Cook, Porter and Carpenter performed what they call distributed ray tracing [Cook1984] to provide a remarkable improvement in antialiasing, along with such effects as motion blur and depth of field, randomly generating rays with slightly different origin, time and intersection point with the screen and then using a weighted average to find the value for a pixel. The jittered distribution was later described more fully by Cook [Cook1986]. Others have since worked to provide improvements in efficiency in generating the sample locations and quality of the distributions, attempting to approach the spectral characteristics of the Poisson disk distribution with algorithms which run nearly as fast as the jittering algorithm [Mitchell1987], [Purgathofer1987], [Lee1985], [Dippé1985].

The problem had been thought to be the presence of high frequency information prior to sampling; it turned out that the problem was the coherence of the aliases produced. Figure 2 shows the demonstration of Yellott. Stochastic sampling as it has been developed since the work of Cook has proven very effective in eliminating aliasing artifacts. This gives us the means to compute images containing no objectionable high frequency information.

Most of the work on antialiasing has assumed that if a good filtered sampling of the image has been found the problem has been solved. The ideal filter, it is assumed, is one which is as narrow as possible (to save computation), and preferably

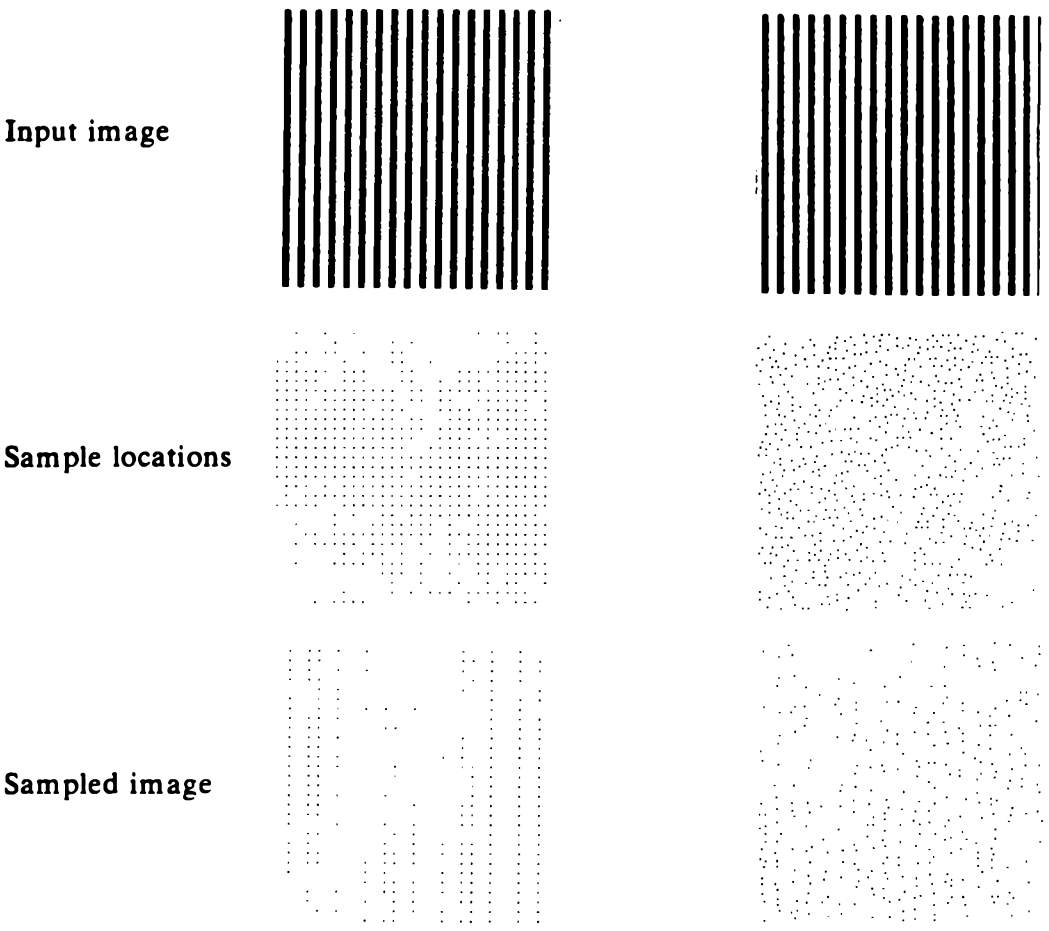

Figure 2. The samples at the right and left have nearly the same average density. On the right, aliasing has been avoided by using a distribution very much like the Poisson disk distribution, as described by Mitchell [Mitchell1987].

non-negative (to avoid having to deal with potentially negative intensity values in the filtered image) while still giving a visually satisfactory result. For example, empirical studies have been done in filter design. Warnock used various choices of filter on high resolution text characters and compared the resulting low resolution versions [Warnock1980]. Crow compared filters on both a mathematically defined pattern and a scene containing long, thin highlights [Crow1981]. He was careful to point out the parameters of his CRT which gave best results, which indicates that he was aware of the device dependence of his results. Mitchell and Netravali undersampled an image and then compared the results of interpolating with various reconstruction filters [Mitchell1988]. All of the comparisons were done visually. The results of these studies depend on the particular display devices being used in the comparison.

Kajiya and Ullner began to address the issue of constructing an image which is tuned for a particular device [Kajiya1981]. The model of display device was a CRT with a regular array of Gaussian pixels. The model of the image was a set of infinitely thin lines, δ functions in one dimension. This was proposed as one model of

an ideal stroke of a character. The Fourier series representation of the image and its construction from Gaussian pixels were derived, and their difference was minimized in a least squares sense. This gave rise to a circulant system of equations which is readily solved for the ideal set of pixel values. Several problems arose, most notably the negative values generated for some pixels. The solution proposed was to use constrained minimization, which produces better results, but at a much greater expense. This is one of the approaches considered in this thesis, although in greater generality. While Kajiya and Ullner considered only Gaussian pixels, and only images consisting of infinitely thin lines, arbitrary shaped pixels and arbitrary images are considered here.

Aliasing, a result of coherent high frequency information, can cause artifacts which are irritating at best, and make the image unrecognizable at worst. Any work in high quality computer imagery must address this problem. Stochastic sampling, by removing the coherence of high frequency information, has been found to be the most effective method to date. Once an image is free from coherent high frequency information, any of the methods of this thesis can be used to produce the best possible realization on an actual raster device. As long as complete information about the image is known (not just intensities at sample points), the methods may be combined with filtering techniques to prevent aliasing artifacts while also preventing artifacts resulting from the structure of the pixels.

Properties of Human Vision

The human visual system detects information only in a specific range of spatial and temporal frequencies. Were this not so, motion pictures would not have the illusion of smooth motion, and computer generated images would have no hope of looking like anything but a collection of coloured dots. In order to use the properties of the visual system to improve the appearance of the images we produce, we need to know something about the way the visual system behaves. This chapter gives a selective overview of those properties which are useful elsewhere in this thesis.

One important concept is that of *contrast*. Contrast is most commonly used to describe an edge. The greater the change in intensity over a short distance, the larger the contrast. Mathematically, contrast is the relative change in intensity at the edge:

$$\text{contrast} = \frac{I_{high} - I_{low}}{I_{high} + I_{low}}.$$

The *dynamic range* of an image is the relative variation in intensity over the entire image; the contrast in the image can be no greater than the dynamic range, but it might be much less.

As discussed in the Appendix, many functions may be represented over a finite interval $0..\Delta$ using the Fourier series

$$f(\xi) = \sum_{n=0}^{\infty} C_n \cos(\varsigma_n \xi + \phi_n) \quad ,$$

where $\varsigma_n = 2n\pi/\Delta$ is the nth angular frequency ($f_n = n/\Delta$ is the nth frequency), ϕ_n is a phase shift applied to the wave at this frequency, and C_n is the amplitude at this frequency. The series is one form of Fourier series or expansion of f. The lowest frequency, $1/\Delta$, is referred to as the *fundamental* frequency, and the next frequency $2/\Delta$, is the *first harmonic*. A pure sinusoidal grating is like a series of equally spaced edges. The contrast at a given frequency is the relative deviation from the mean caused by fluctuations in image intensity at that frequency:

$$\text{contrast} = \frac{(C_0+C_n)-(C_0-C_n)}{(C_0+C_n)+(C_0-C_n)} = \frac{C_n}{C_0}.$$

Since negative intensities are not possible, C_0 must be greater than 0, unless all of the amplitudes are 0. The contrast threshold at a given frequency is the minimum contrast for which (spatial or temporal) sinusoidal variations in intensity are detectable.

Linearity in the human visual system, to the extent that it may be assumed, simplifies the process of modelling the system. When the input to a linear system is a sinusoid, the output is changed only in amplitude and phase (no new frequencies are introduced). The response of a linear system to the sum of two inputs is the sum of its responses to the two inputs. To a considerable extent, the human visual system can be considered linear; there is psychophysical evidence to support this [Kaufman1974]. On the other hand, there is physiological evidence for non-linearity. If the human visual system were truly linear, then no stimulus could ever be totally invisible, or else the same stimulus at a higher amplitude would still be invisible. Any continuous function behaves in a linear fashion for small deviations (e.g. from a mean value of the argument). By Taylor's theorem, if $F(I)$ gives the response to an intensity I, then for small contrast at frequency C_n,

$$f(C_0+C_n) = f(C_0) + C_n \partial \frac{f(C_0)}{\partial C_n} + \cdots$$

This means for small amounts of contrast (near the threshold of visibility), the response is nearly linear with respect to single frequencies. In high quality computer generated images the emphasis is on removing small deviations from the ideal. The contrast of the deviations must be small or else the image is not even close to ideal. This means that the assumption of linearity is reasonable, for single frequencies. Where multiple frequencies are involved, cross terms must be introduced into the series, so that the linear term is not so simple. Thus the coefficient of C_n is not a constant, or even a function of C_n alone, but depends on the contrasts at other frequencies as well. Unfortunately the interrelationships are not yet available in the vision literature, so we have assumed a simpler model (which ignores such cross terms).

There is good experimental evidence that the system is linear for small increments, even for multiple frequencies. For example, a complex waveform made up of the sum of several sinusoidal components may be invisible unless one of the components would be visible on its own. Where two components have sufficiently different frequencies they are detected independently, but frequencies which are close together are detected as one. Campbell and Robson [Campbell1968] measured the visibility of gratings with intensity profiles which were sinusoidal, square-wave,

rectangular-wave, and saw-tooth. Figure 3 shows a representative graph from their work. In this case, a high contrast square wave grating is compared with a sinusoidal grating whose frequency is the same as the fundamental frequency of the square wave.

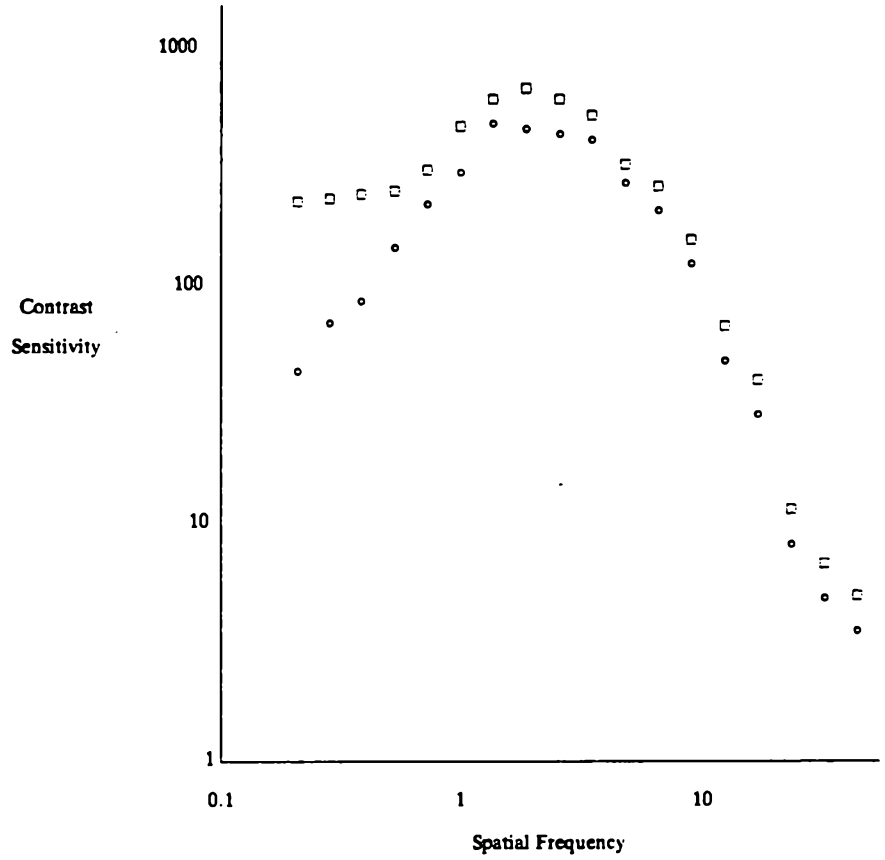

Figure 3: Contrast sensitivity as a function of spatial frequency. Square wave grating: □ Sinusoidal grating: ∘. At the low spatial frequency end, the first harmonic is more visible than the fundamental, and so the sensitivity is increased for the square wave over what it would be for the fundamental alone. From [Campbell1968].

Over a wide range of spatial frequencies, the contrast threshold of any of these complex waves is entirely determined by the amplitude of the fundamental component of its Fourier expansion. Gratings of complex wave forms cannot be distinguished from sinusoidal until the contrast is above the contrast threshold for the next harmonic.

Blakemore and Campbell found separate spatial channels with a bandwidth of just over an octave [Blakemore1969], while Sachs, Nachmias and Robson found that different frequencies are detected independently if the frequencies differ by 20%

[Sachs1971]. This suggests that it is a good first approximation to treat errors at different frequencies as if they are detectable independently, although a better approximation would group neighbouring frequencies together. Wilson and Gelb found exactly six slightly overlapping channels [Wilson1984], however Nielson and Wandell pointed out that the Wilson-Gelb model on its own considers far too many images indistinguishable [Nielson1988]. The visual system is clearly not as simple as a linear system. At this point, the better models put forward in vision research are still in dispute. Using a linear model, at least for small deviations (near threshold), is better than ignoring the visual system, and it is the approach followed in this thesis.

The effect of the various phase shifts in the series is sometimes more important than amplitude. Oppenheim demonstrated [Oppenheim1981], at least for some images, that if the set of amplitudes from two images are interchanged, while retaining the phase shifts, the resulting images are clearly recognizable from the phase information alone. A simple example which demonstrates the importance of phase in at least one pair of images is given in Figure 4. It has been argued by some that many natural images may be well described statistically by a $1/f$ distribution [Mandelbrot1977]. The two fractal curves of Figure 4 were constructed to have an approximate $1/f$ distribution, with uniform, but different random phase. In the lower half of the figure, the phases have been interchanged. The images which share phase information are more similar than those which share frequency information, suggesting that for this example, frequency information is less important than phase information.

There are several forms of Fourier series, one of which is obtained by expanding the terms as

$$C_n \cos(\omega_n \xi + \phi_n) = C_n(\cos(\phi_n)\cos(\omega_n \xi) - \sin(\phi_n)\sin(\omega_n \xi))$$

$$= A_n \cos(\omega_n \xi) + B_n \sin(\omega_n \xi).$$

In this formulation, if the values of A_n and B_n are correct, then the phases are as well, and small errors in these values appear as small errors in the phase. Larger errors lead to errors in phase which are not linearly related to the errors in the values of A_n and B_n, but in high quality images, such large errors should not exist. Most data available in the vision literature measures frequency response, and not phase response, although errors in phase are important. This is at least partially corrected by using this form of series.

Spatial frequency response measurements give the threshold contrast level for visibility of an achromatic sinusoidal grating as a function of spatial frequency (response to chromatic contrast is more complicated; we will limit ourselves to the achromatic case). Results of spatial frequency response measurements are shown as

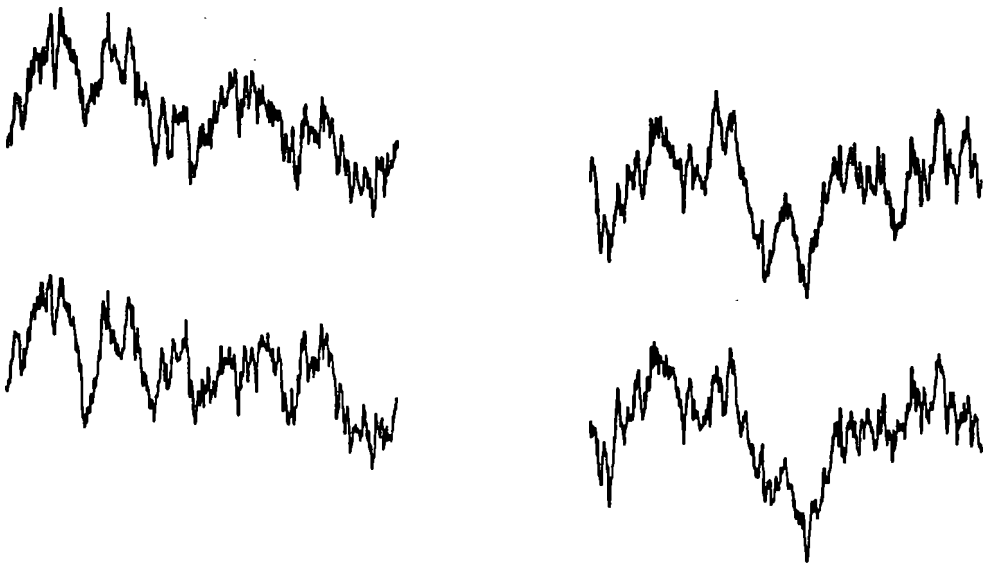

Figure 4: At top are two randomly generated fractal curves. These would be appropriate as one-dimensional intensity profiles of some varieties of clouds. On the bottom left, the phase information from the top left curve has been combined with the amplitude information from the top right curve. On the bottom right, the reverse pair is shown.

graphs of *contrast sensitivity*: the inverse of the threshold contrast for visibility. Qualitatively speaking, the spatial response of the human visual system is like a band-pass filter. Very high frequency variations are invisible, and we are more sensitive to mid-range frequencies than to low frequencies (see lower curve of Figure 3). For images which vary with time, the spatial contrast threshold depends on the temporal frequency. Figure 5 shows the spatial frequency response for several temporal frequencies, as measured by Robson. The 1 Hz curve is much like the one for static images in Figure 3.

The analogue in the temporal domain to grating measurements in the spatial domain is a flicker sensitivity measurement. Sensitivity is highest at low frequencies, being about level up to a cutoff frequency, and then drops fairly quickly as frequency increases past about 10 Hz. The width and position of the peak, if any, is quite sensitive to the spatial properties of the image. Figure 6, from the same Robson paper as Figure 5, shows the threshold temporal contrast at four spatial frequencies [Robson1966]. An approximate spatiotemporal threshold surface may be constructed by combining the data from Figures 5 and 6. This is approximated by a single bicubic patch in Figure 7.

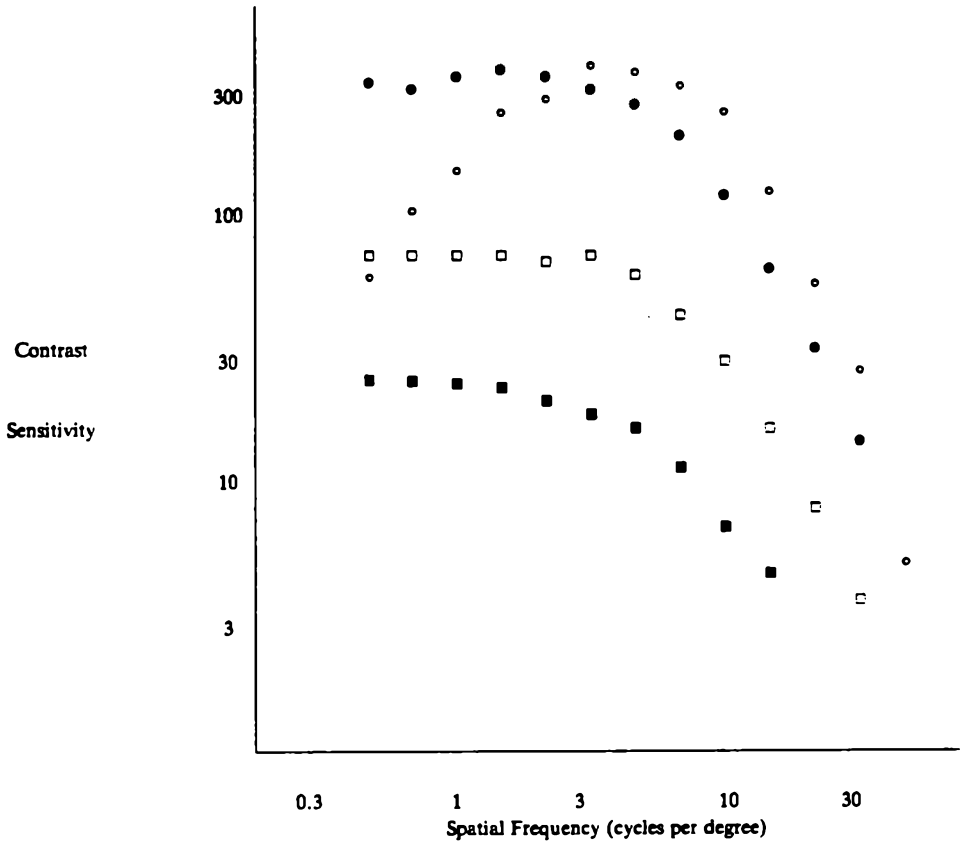

Figure 5: Spatial contrast threshold at various temporal frequencies. ∘ 1, ● 6,□ 16,■ 22 Hz. From [Robson1966].

Under computer control, it is possible to correct for eye movements. Kelly performed a series of measurements to determine the effect of retinal image motion on perception. In them input from an eye tracker was used to move the image on the raster, thereby eliminating changes in the image due to eye movements [Kelly1979]. From the response to moving gratings, he was able to construct the spatiotemporal threshold surface [Kelly1979a]. Like the surface of Figure 7, the surface he found was not separable into the product of a spatial response function and a temporal response function, since the shape of the spatial frequency response curves change at low temporal frequencies.

Kelly also measured thresholds for periodic spatial patterns containing two or more differently oriented components [Kelly1982]. In unstabilized images, each component is detected independently. If image position is corrected for eye movements, the independence goes away. Since normal viewing conditions do not involve stabilized images, this loss of independence is not relevant to most computer

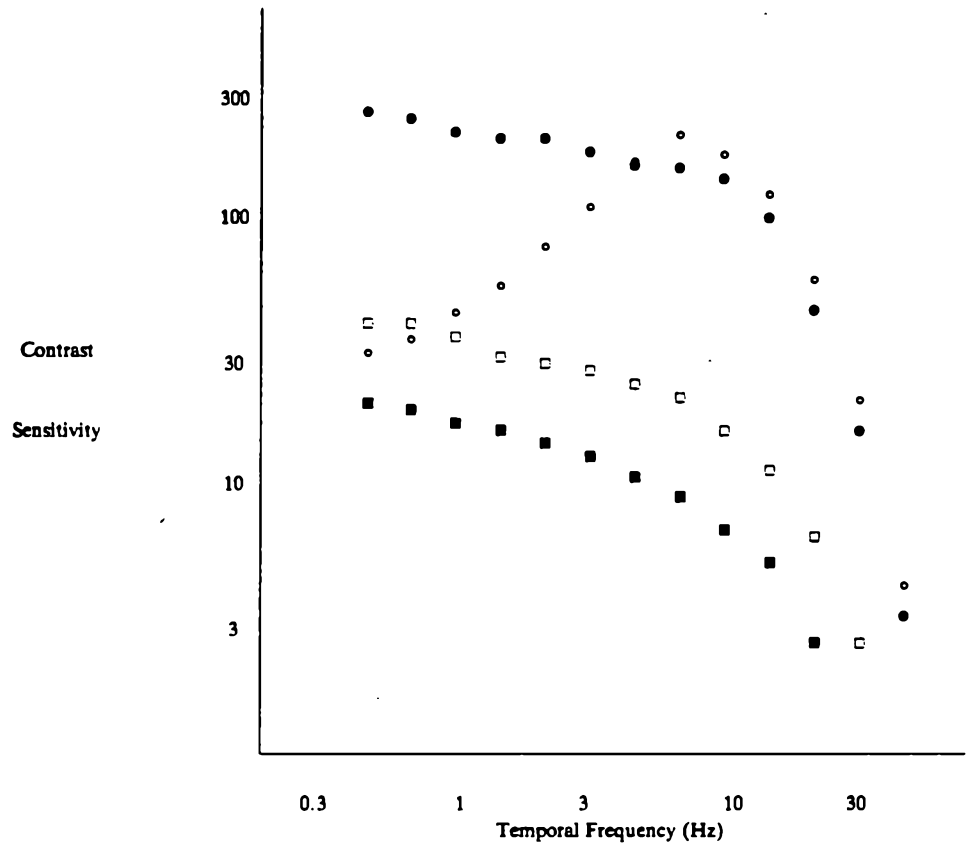

Figure 6: Temporal contrast threshold at various spatial frequencies ∘ 0.5, • 4, □ 16, ■ 22 cycles per degree. From [Robson1966].

image generation. This is further evidence in favour of a linear model, when the image is not fixed to one position on the retina.

Koonderink and van Doorin [Koonderink1979] repeated Kelly's experiments without stabilization [Kelly1979a], and found the bi-modal surface shown as a contour plot in Figure 8. One peak of sensitivity is where previous results predicted (0.5 cycle/degree,7.6 Hz); another is at approximately 4 cycle/degree, 0 Hz. Once again the surface is not separable.

All three of these surfaces (Kelly's, Koonderink and van Doorin's, and the one constructed from Robson's data) are alike to first order. For computational expediency, Watson *et al.* proposed a "window of visibility" approach, based on the high frequency cutoffs in both the spatial and temporal domains [Watson1986]. For the sake of computer generated animation, they suggest assuming that anything outside of this window (which corresponds roughly to the outer contours of Koonderink and van Doorin's diagram) is invisible, and treating everything within the

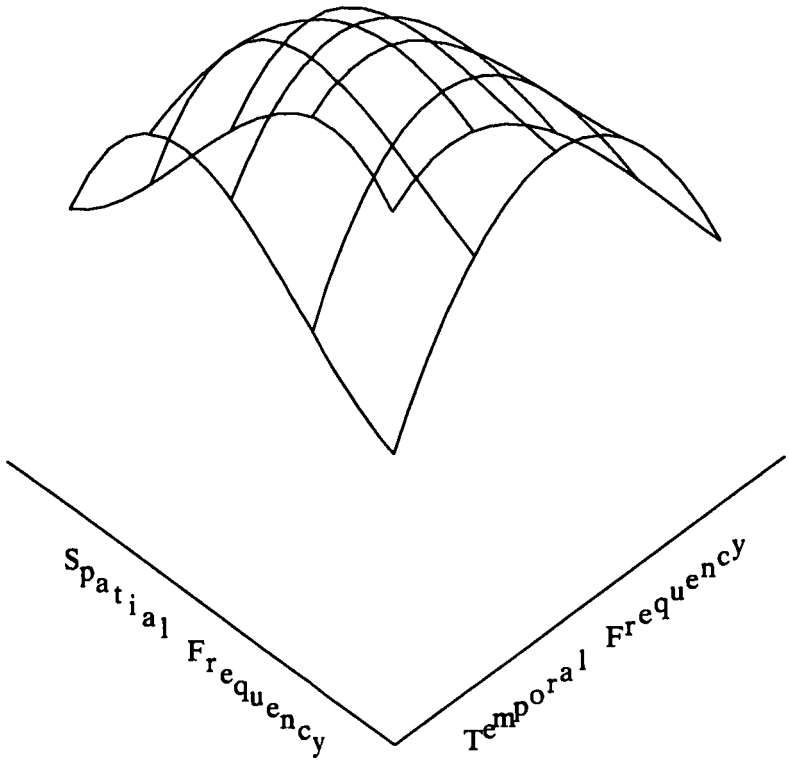

Figure 7: An approximation of the spatiotemporal threshold surface. A single bicubic patch was fit by the method of least squares [Forsey1990] to the data of Figures 5 and 6. Curves of constant spatial and temporal frequency are drawn along the surface.

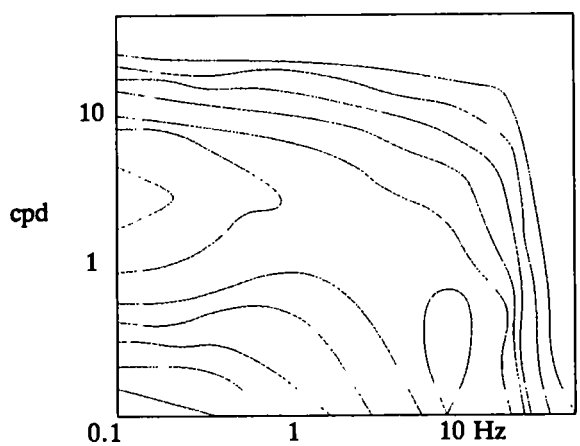

Figure 8: Contour plot of the spatio-temporal contrast threshold surface. After [Koonderink1979].

window as equally important, performing antialiasing if this leaves frequencies above the high frequency cutoff for the display.

In this thesis, the visual system is characterized purely on the basis of its frequency response. This is only an approximation; the human visual system is not linear, and its sensitivity depends on more than the frequency content of stimuli. Vernier acuity experiments measure the ability of the visual system to discriminate small deviations from straightness [Thomas1975], [Fahle1981]. A thin vertical line is broken at the middle, and the top half displaced horizontally by an amount which regular acuity measurements and spatial frequency measurements would indicate should be invisible, and yet the displacement can be seen. The frequency distributions of the top and bottom pieces are identical, but all of the components have a constant phase shift which is being detected. Other effects which cannot be explained purely on the basis of frequency response are adaptation [Barlow1958], [Connor1982] and motion sensitivity [Braddick1974], [Westheimer1975], [Burr1979]. The difficulty which arises when trying to use a more sophisticated model of the visual system is that the appropriateness of the various models is still under debate in the vision literature. The spatiotemporal surface gives a good estimate of what is not visible. A more exact estimate of levels of visibility must wait until the models accepted by the vision research community are more complete.

Problem Definition

Practical display devices are approximations to ideal devices. These ideal devices have display surfaces consisting of identical pixels aligned on a rectangular grid. The formalism in this thesis is designed for such devices. Not all devices fit this description. For example, there exist experimental devices with scanlines which are horizontally offset from one another [Wittke1987]. With large scale integration, liquid crystal displays could be manufactured with pixels at random locations. It is possible to extend the formalism to include such cases, but the examples given in this thesis do not use such extensions.

There are many possible descriptions of an image, including a continuous mathematical function, a regularly spaced sampling, and a random sampling. Of the possible descriptions, several representative classes of image specification are explored in detail in this thesis. This chapter gives the assumptions about devices which are used in later chapters, and the classes of image specifications which are considered.

A displayed image is specified by giving a sum of intensity values (scalars) multiplied by pixel intensity profiles (functions of space and time). In general the pixel intensity profiles need not be translations of a single common profile, but in the examples presented in this thesis they are. Pixels are assumed to sum independently and behave in a linear fashion; the intensity value due to one pixel is independent of intensity values at other pixels, and proportional to the value at that pixel. The intensity of the image at any point is given by a sum

$$\sum_j I_j B_j(x, y, t) \quad ,$$

where j runs over all pixels. I_j is the intensity value for the jth pixel, and $B_j(x, y, t)$ is the pixel intensity function of the jth pixel. The set of all images which may be described in this way forms a vector space; one of the properties which the pixel intensity functions are assumed to have is that they form a basis for this space. For this reason they are called pixel basis functions or simply basis functions. The set of images realizable on a device is a subspace of this vector space; no negative intensities

or intensities greater than the device limits may occur in a realizable image. When the pixel basis functions are translations of a single function, they are written $B(x-x_j, y-y_j, t-t_j)$.

The specific assumptions about devices are the following.

1) It is assumed that a device consists of a regular array of pixels and an array of weights which are applied to the pixels to provide an intensity variation. For the purposes of this thesis, the array of weights is called the *frame buffer*.

2) The pixel intensity distributions are translations of a single pixel basis function.

3) The pixel basis must tend to zero as the distance (in space and time) from the pixel origin goes to infinity, and it must be normalized, so that an infinite array of pixels with unit weight has unit integrated intensity over unit distance. The intensity at ξ is given by

$$\sum_{p=-\infty}^{\infty} B(\xi - p\,\delta).$$

The integrated intensity over unit distance is the integrated intensity between any two pixels divided by the interpixel separation, δ, so the normalization condition is

$$1 = \frac{1}{\delta}\int_0^{\delta} \sum_{p=-\infty}^{\infty} B(\xi - p\,\delta)\,d\xi \tag{1}$$

$$= \sum_{p=-\infty}^{\infty} \frac{1}{\delta}\int_0^{\delta} B(\xi - p\,\delta)\,d\xi \tag{2}$$

$$= \frac{1}{\delta}\int_{-\infty}^{\infty} B(\xi)\,d\xi. \tag{3}$$

Equation (3) follows from Equation (2) because adjacent elements of the sum in Equation (2) are integrals of the same function, over neighbouring segments.

A one dimensional image produced by an infinite raster device with the vector **a** containing the raster values is then

$$I(\xi) = \sum_{p=-\infty}^{\infty} a_p B(\xi - p\,\delta) \quad ,$$

with ξ one of x, y, or t. If $\xi \equiv t$ represents time, then the weight a_p is applied at time $t_p = p\,\delta$, corresponding to the pth frame in an infinite sequence. As an example, an ideal CRT being driven from an $n \times m$ frame buffer running at f Hz for T seconds

would be modelled as an array of $n \times m \times fT$ pixels, each of which is a translation in space and time of the function

$$B(x,y,t) = \begin{cases} \lambda\tau \frac{\delta^2}{\sigma^2 2\pi} e^{-\lambda t} e^{-(x^2+y^2)/2\sigma^2}, & t \geq 0 \\ 0, & t < 0. \end{cases}$$

At any time, such a pixel has a Gaussian intensity distribution in space, and the intensity dies off exponentially with time, at any point in space. Here τ is the inter-frame interval, δ is the inter-pixel separation (in x and y, assuming the pixel spacing is the same in x as in y), λ is the phosphor decay constant, and σ is the half-width of the spatial intensity profile. More generally, the half-width and inter-pixel separation can be specified separately in x and y.

The example of a CRT is frequently used to demonstrate the methods developed. This is partially because of the familiarity of this device, and its availability for testing. The methods are intended to be more generally applicable, and should not be construed as CRT specific. The LCD, for example, does not switch instantly. Whatever switching characteristic it has can be analysed by the methods described below.

Two special cases which come up often are the static image and the flat field. Many computer graphics images are static (animated ones are still the exception rather than the rule), so this case is directly useful in practice. The flat field is less interesting to computer graphics but permits experimentation in a one-dimensional temporal domain. When either of these is being used, the unchanging variables (either x and y for a flat field or t for a static image) are omitted.

In the flat field case, the problem is to find a set of weights that provides the nearest temporal image intensity variation to $I(t)$. (There are various definitions of *nearest* which are explored later. Ideally, it means "most similar visually", but this is usually approximated.) More specifically, it is the problem of finding a set of weights a_p such that

$$J(t) = \sum_{p=-\infty}^{\infty} a_p B(t - t_p)$$

is as close as possible to $I(t)$. For example, suppose that the basis function describes the exponential decay of CRT phosphors, as above. Then the intensity of a pixel at any time depends on the intensity of the pixel at every frame in the past. Each intensity is reduced by the exponential decay of the phosphors since that frame. The

expression above is for arbitrary pixel spacings. For uniformly spaced pixels, it would be written as

$$J(t) = \sum_{p=-\infty}^{\infty} a_p B(t - p\tau) \quad .$$

The ideal image $I(t)$ may be specified in a variety of forms. The only restriction is that it must not contain coherent information at frequencies above the limits reproducible by the device. The simplest form for discussion is that of a continuous closed form function. In practice, this is seldom available. More common forms specify the intensity at some set of locations, and may or may not include information about what the image does between specified locations. This is the sort of image calculated by most sampling methods, including stochastic sampling. The other common form gives an approximation of the average of the image between specified locations. This form of image specification is provided by techniques which use filtering, or approximate it using weighted averages of samples in a region.

One representation which is seldom considered in this way is that of an image which is specified as a linear combination of weighted basis functions. A specific instance of this is an image which has been computed or otherwise obtained for display on a CRT with a frame buffer of resolution $n_1 \times m_1$, such as an image which is hand-crafted using a computer paint program. Finding the best possible set of weights for a different resolution device (say, $n_2 \times m_2$) with a possibly different basis function (perhaps a printer, or an LCD display) is a special case of the problem under discussion, but a very important one. For monochrome images the solution to this special case completely specifies how to move an image from one device to another. It may be that in a device-to-device transfer some filtering is required to remove high frequencies before the solution may be applied.

Another special case involving change of resolution and change of basis function arises in computing for an excessively high resolution device. The technology of high resolution devices is steadily advancing. Already there exist devices with resolutions exceeding the limits of the human visual system. If the high resolution device is the target device from the start, then one method of saving computation time is to compute the image for a lower resolution device with a carefully chosen pixel profile and then change the basis and resolution to match the high resolution device, in effect building bigger pixels with "nice" basis functions out of smaller ones with given profiles. The question to be answered is then one of which pixel profile is best.

One important detail has been omitted in specifying the problem. It is how the difference between the ideal image and an approximate solution is measured. Several approaches are available, each having its advantages and drawbacks.

1) Sample the image once per pixel. Minimize the difference between the image desired and the image constructed from weighted basis functions at the sample points. If the image is available only at these sample points, this approach is the only choice. This approach is easiest, and under some conditions can be done exactly (an example of such conditions is discussed in Section 8.1). In other cases, a norm of the difference vector is minimized.

2) Sample the image more than once per pixel, and minimize differences at sample points as in approach 1). This approach might be contemplated for images which are supersampled for purposes of antialiasing. Since the shape of the basis function is fixed, only one parameter can be adjusted per pixel. This means that the problem is overdetermined, and an exact solution is unlikely to exist. If there is any information to be gained by considering what happens between pixels, then Whittaker's theorem indicates that the image contains frequencies above the limits of what the display can reproduce. Hence the image should be filtered and then sampled at the lower rate, and method 1) used.

3) Define a function localized around sample points, and use it to find weighted averages of the image. Minimize differences between weighted averages and the image produced on the display at corresponding points. An example of such a function is the box function. It can be used to lower high frequency content. With the box function the total intensity of the image is preserved, and the average intensity in any neighbourhood comes close to matching that of the ideal image in the corresponding neighbourhood. If the integral of the ideal intensity function is known the computation is similar in difficulty to 1). If the ideal image is known only at a set of sample points, the points may be interpolated using a polynomial function, giving a new intensity function, which may be integrated. This is tantamount to numerical integration.

4) Use non-localized functions, such as Fourier transforms. For example, minimize the difference between the coefficients of the Fourier expansions of the ideal and approximate images. The coefficients can be weighted according to their visual importance, using the relative visibility of information at the corresponding frequencies. This method has several advantages. First, if the image contains unwanted high frequency information, it can be eliminated when the Fourier expansion is calculated. Secondly, this method can use the properties of the human visual system to improve the quality of the result. Its disadvantage is that images are rarely provided as Fourier expansions and so a transformation is required to find the Fourier expansion of the image.

In this thesis, method 4 using Fourier expansions is referred to as *frequency space minimization*, while method 1 is called *direct minimization*. Methods 2 and 3 are variants on the direct minimization approach.

With infinite computational resources, and arbitrarily fine sampling density, one can get an arbitrarily close match with respect to any metric. When the sampling density is finite, such perfection is no longer available. In this case the choice of metric matters. The metric of choice is a visual metric, that is, the best method must produce an image as visually close to the ideal image as is possible given the finite sampling density. This is not normally an available option, but of the methods available, we choose the one which approximates this as nearly as any. Where different methods give visually equivalent results, the most computationally efficient one is the method of choice.

It is often convenient to assume that the ideal image is periodic. Periodicity introduces no error into the solution as long as the period is sufficiently long. The basis functions are guaranteed to decay to values less than machine precision by a finite distance from their origin. Let the period be larger than the image size or duration by this distance. Then the neighbouring copies of the image are sufficiently far away that their existence has no effect. In a similar way, the assumption of periodicity does not prevent local computation. If the reconstruction of only a part of an image is desired, enough of the neighbourhood of the reproduced part should be included to prevent the periodicity from being visible.

A photographer, by changing camera settings, filters or film, can accommodate most intensity ranges and produce a photograph with much the same tonal range. In the same way, in computer graphics it is considered fair to scale the intensity of an image to the 0..1 range or any other range that is convenient. Given an image in the 0..1 range, it might still be the case that the best match to that image on a given device requires values for some pixels outside the range of the device. There are two solutions possible. One is to find the best match subject to the constraints that the values lie within the 0..1 range. This involves constrained optimization. The other is to adjust all images before solving, so that an image initially in the 0..1 range is adjusted to be in a smaller range, which then produces a set of frame buffer values within the 0..1 range. This involves a reduction of dynamic range. A contrast reduction should depend on the device and not on the image, so that dim images remain dim, and bright images remain bright. A moderate reduction of contrast is relatively unimportant. A major reduction results in an unsatisfactory image.

In summary there are three choices to be made: Is the minimization in the frequency domain or the temporal/spatial domain? Is the contrast adjusted overall or is the minimization performed with constraints? With respect to which norm is the

minimization calculated? The next two chapters make it possible to select either of the answers to the first two questions, and any of three answers to the third. The following chapters discuss tradeoffs between the choices available.

Mathematical Tools and Algorithms

To proceed further requires the use of a number of mathematical tools. Each of the methods of solution described can be cast in the form of the minimization of the difference between two vectors (sets of values), in which one is the product of a matrix and a vector. While some knowledge of matrix algebra is assumed, a sufficient set of definitions is given to establish the notation, for which no universal conventions exist. Circulant matrices play an important role in several forms of the minimization. Because they are often considered too obscure or specialized to be treated in texts on linear algebra, they are defined, and several theorems relating to them are given. Several ways in which they arise are discussed. A related new form, termed here *pseudo-circulant*, is also presented, along with ways in which they arise (which, not surprisingly, are closely related to ways in which circulant matrices arise).

The theorems presented here are given to provide a basis for what follows, not to provide a balanced understanding of circulant matrices. Circulant matrices are closely related to Fourier series. The Appendix contains a number of useful results about Fourier series and transforms which are used in this chapter.

The first section presents a way of describing a raster image using its Fourier series. For images which are periodic in space or time, the Fourier series representation is of the form of a sum of weighted Fourier transforms of the pixel basis function. Section 5.2 establishes the notation used throughout this thesis for matrices and some operations applied to them, as well as defining certain special forms. Section 5.3 introduces circulant matrices, and shows how they arise in this thesis. Section 5.4 extends circulant matrices to a new form, which are in a sense the rectangular analogue to circulant matrices (which are by definition square).

5.1. Fourier Series Representation of an Image

An image defined as a linear combination of pixels has an intensity at a point which is in principle an infinite sum. (The pixel basis may make the sum finite). If the image is periodic, or finite in extent, it can be described in terms of its Fourier series. This allows us to compare the terms of the series of the image produced by the device with those of the desired image. To make a finite image periodic it is necessary to extend it far enough that the pixels used to display one period have no effective contribution at the next period. The following theorem gives the Fourier series for a periodic function made up of pixels.

THEOREM 1: *If* $f(\xi)=\sum_{p=-\infty}^{\infty} w_p G(\xi-p\delta)$ *and* f *is periodic with period* $\Delta=N\delta$, *with* $\lim_{\xi\to\pm\infty} G(\xi)=0$, *and* $\int_0^{\delta} G(\xi)d\xi > 0$, *then*

1) $w_p = w_{p+N}$,

2) $$f(\xi)=a_0+\sum_{n=1}^{\infty} a_n \cos\varsigma_n\xi + b_n \sin\varsigma_n\xi,$$

with $$a_0=\frac{\sqrt{2\pi}}{\Delta}\sum_{l=1}^{N} w_l \mathcal{F}_c[G(\xi-l\delta),0]\ ,$$

$$a_n=\frac{2\sqrt{2\pi}}{\Delta}\sum_{l=1}^{N} w_l \mathcal{F}_c[G(\xi-l\delta),\varsigma_n]\ ,$$

and $$b_n=\frac{2\sqrt{2\pi}}{\Delta}\sum_{l=1}^{N} w_l \mathcal{F}_s[G(\xi-l\delta),\varsigma_n]\ .$$

Here, $\mathcal{F}_c[G(\xi),\varsigma]$ is the Fourier cosine transform of the function $G(\xi)$, evaluated at frequency ς, and $\mathcal{F}_s[G(\xi-l\delta),\varsigma_n]$ is the Fourier sine transform. See the Appendix for details.

Proof:

1) From the periodicity of the image,

$$\sum_{p=-\infty}^{\infty} w_p G(\xi - p\,\delta) = \sum_{p=-\infty}^{\infty} w_p G(\xi - (p+N)\,\delta) \quad .$$

Regrouping the sum,

$$\sum_{p=-\infty}^{\infty} w_p G(\xi - p\,\delta) = \sum_{q=-\infty}^{\infty} \sum_{l=1}^{N} w_{qN+l} G(\xi - ((q+1)N + l)\,\delta).$$

If the same regrouping is done with the left hand side, the equation becomes

$$\sum_{q=-\infty}^{\infty} \sum_{l=1}^{N} w_{qN+l} G(\xi - (qN + l)\delta) = \sum_{q=-\infty}^{\infty} \sum_{l=1}^{N} w_{qN+l} G(\xi - ((q+1)N + l)\delta). \qquad (4)$$

Substituting $r = q + 1$, on the right hand side,

$$\sum_{q=-\infty}^{\infty} \sum_{l=1}^{N} w_{qN+l} G(\xi - (qN + l)\,\delta) = \sum_{r=-\infty}^{\infty} \sum_{l=1}^{N} w_{(r-1)N+l} G(\xi - (rN + l)\,\delta) \quad .$$

Since $G(\xi)$ has a finite integral but tends to zero at large values of ξ, $G(\xi - (qN + l)\,\delta) \equiv G(\xi - (rN + l)\,\delta)$ can only hold for all values of ξ if $q = r$. Expressing both sides with the same indices,

$$\sum_{q=-\infty}^{\infty} \sum_{l=1}^{N} w_{qN+l} G(\xi - (qN + l)\,\delta) = \sum_{q=-\infty}^{\infty} \sum_{l=1}^{N} w_{(q-1)N+l} G(\xi - (qN + l)\,\delta).$$

This equation is an identity with respect to ξ, so the coefficients w_q must be equal. Hence $w_{qN+l} = w_{(q-1)N+l}$, or $w_{qN} = w_{(q-1)N}$. Since this holds for arbitrary q, $w_p = w_{p+jN}$.

2) Using the periodicity of w_q in the regrouped form of f (4),

$$f(\xi)=\sum_{j=-\infty}^{\infty}\sum_{l=1}^{N}w_l G(\xi-(jN+l)\delta)=\sum_{l=1}^{N}w_l\sum_{j=-\infty}^{\infty}G(\xi-(jN+l)\delta)\ ,$$

$$\text{so } a_0=\frac{1}{\Delta}\sum_{l=1}^{N}w_l\sum_{j=-\infty}^{\infty}\int_0^{\Delta}G(\xi-(jN+l)\delta)dt=\frac{1}{\Delta}\sum_{l=1}^{N}w_l\int_{-\infty}^{\infty}G(\xi-l\delta)dt$$

$$=\frac{\sqrt{2\pi}}{\Delta}\sum_{l=1}^{N}w_l\mathcal{F}_c[G(\xi-l\delta),0]dt\ ,$$

by the definition of Fourier series.

$$\text{Similarly } a_n=\frac{2}{\Delta}\sum_{l=1}^{N}w_l\int_{-\infty}^{\infty}\cos\zeta_n\xi G(\xi-l\delta)dt=\frac{2\sqrt{2\pi}}{\Delta}\sum_{l=1}^{N}w_l\mathcal{F}_c[G(\xi-l\delta),\zeta_n]dt\ .$$

The equation for b_n follows similarly. □

The two dimensional extension follows, using the complex exponential form for conciseness.

THEOREM 2: *If* $f(x,t)=\sum_{p=-\infty}^{\infty}\sum_{q=-\infty}^{\infty}w_{pq}B_x(x-q\kappa)B_t(t-p\tau)$, *with* f *periodic in both* x *and* t *with periods* D *and* T *respectively, and* B_x *and* B_t *both normalized as before, then* $f(x,t)=\sum_{r,s=-\infty}^{\infty}z_{rs}e^{i(k_r x+\omega_s t)}$,

with $z_{rs}\sum_{i,j}w_{ij}\frac{2\pi}{4DT}\mathcal{F}[B_t(t-j\tau),-\omega_s]\,\mathcal{F}[B_x(x-i\kappa),-k_r]$.

Proof: The proof is similar to the one dimensional proof, except that the complex exponential form is used:

$$f(x,t)=\sum_p\sum_q w_{pq}B_x(x-q\kappa)B_t(t-p\tau)$$

$$=\sum_{r=-\infty}^{\infty}\sum_{i=1}^{M}\sum_{s=-\infty}^{\infty}\sum_{j=1}^{N}w_{ij}B_x(x-(rM+i)\kappa)B_t(t-(sN+j)\tau)\ ,$$

with $M\kappa=D$ and $N\tau=T$.

So

$$z_{mn} = \frac{1}{4DT}\int_{-D}^{D}\int_{-T}^{T}\sum_{r,i,s,j} w_{ij}B_x B_t\, e^{-i(k_r x+\omega_s t)}dx\,dt$$

$$= \frac{1}{4DT}\sum_{i,j} w_{ij}\sum_{r=-\infty}^{\infty}\int_{-D}^{D}\sum_{s=-\infty}^{\infty}\int_{-T}^{T} B_t(t-(sN+j)\tau)e^{-i\omega_s t}B_x(x-(rM+i)\kappa)e^{-ik_r x}dx\,dt$$

$$= \sum_{i,j} w_{ij}\frac{1}{4DT}\int_{-\infty}^{\infty} B_t(t-j\tau)e^{-i\omega_s t}\int_{-\infty}^{\infty} B_x(x-i\kappa)e^{-ik_r x}dx\,dt$$

$$= \sum_{i,j} w_{ij}\frac{2\pi}{4DT}\,\mathcal{F}[B_t(t-j\tau),-\omega_s]\,\mathcal{F}[B_x(x-i\kappa),-k_r]\ .$$

□

5.2. Matrix Notation

A variety of different matrix notations are in current use. The notation used in this thesis is chosen for its mnemonic value.

DEFINITION 1: *(Matrix notation). A* **matrix M**, *of* **order** $m \times n$, *is an array of numbers, consisting of m rows and n columns. The element of* **M** *in row i and column j is denoted* m_{ij}. *Where the row or column index is an expression (such as* $i+1$*), the two indices are separated by a comma: e.g.* $m_{i+1,j-2}$. *A matrix having only one row or one column is called a* **row-** *or* **column-vector**: *a column vector of order n is the same as an array of order* $n \times 1$.

DEFINITION 2: *(Vector norm). The norm of a vector* **v**, *denoted* $\|\mathbf{v}\|$, *is a measure of the size of the vector. In general, the* l_k *norm of* **v** *is defined as*

$$\|\mathbf{v}\|_k = \left(\sum_i |v_i|^k\right)^{1/k}\ .$$

The three most commonly used norms are the l_1 norm, which is the sum of the absolute values of the elements of **v**, the l_2 norm, which is the square root of the sum of the squares of the elements, and the l_∞ norm, which is the magnitude of the largest element of **v**.

DEFINITION 3: *(Matrix multiplication). If* **A** *is an* $l \times m$ *matrix and* **B** *is an* $m \times n$ *matrix, then the* $l \times n$ *matrix* $\mathbf{C} = \mathbf{AB}$ *is the (matrix) product of* **A** *and* **B**. *It has elements* $c_{ij} = \sum_{l=1}^{m} a_{il} b_{lj}$.

A common problem in matrix algebra is to find the vector **x** which satisfies $\mathbf{Ax} = \mathbf{b}$. If

$\mathbf{A}$ is square and full rank, then this can be solved; otherwise, the norm of the vector $\mathbf{Ax}-\mathbf{b}$ is minimized. The error can be characterized by the single number which is the value of the norm, or by the entire vector.

DEFINITION 4: *(Residual vector) In a matrix vector problem of the form* minimize $\|\mathbf{Ax}-\mathbf{b}\|$, *possibly subject to some constraints, the vector* $\mathbf{Ax}-\mathbf{b}$ *containing the errors corresponding to each of the elements of* $\mathbf{b}$ *is the* **residual vector**.
Note that the elements of the residual vector do not directly correspond to errors in the elements of the result. They only give an indication of how well the elements of the right hand side vector are approximated by the product $\mathbf{Ax}$.

DEFINITION 5: *(Complex conjugate). If* $\mathbf{Z}$ *is a complex matrix, it may be expressed as the sum of a real and an imaginary matrix,* $\mathbf{A}+i\mathbf{B}$, *with elements* $z_{ij}=a_{ij}+ib_{ij}$. *The* **complex conjugate** *matrix of* $\mathbf{Z}$, *denoted* $\mathbf{Z}^*$ *is then* $\mathbf{A}-i\mathbf{B}$.
In this thesis, the only complex matrices appear in the frequency space formulation.

DEFINITION 6: *(Matrix transposition). If* $\mathbf{M}$ *is an* $m\times n$ *matrix, then the transpose of* $\mathbf{M}$, *written* $\mathbf{M}^T$, *is an* $n\times m$ *matrix with* $m_{ij}=m_{ji}^T$.

DEFINITION 7: *(Hermitian transpose). If* $\mathbf{M}$ *is an* $m\times n$ *matrix, then the Hermitian transpose of* $\mathbf{M}$, *written* $\mathbf{M}^H$, *is* $(\mathbf{M}^T)^*=(\mathbf{M}^*)^T$.

DEFINITION 8: *(Matrix inverse). The* **inverse** *matrix of an* $n\times n$ *matrix* $\mathbf{M}$, *denoted* $\mathbf{M}^{-1}$ *is the* $n\times n$ *matrix for which* $\mathbf{MM}^{-1}=\mathbf{I}$, *where* $\mathbf{I}$ *is the identity matrix, which has ones along the principal diagonal and zeroes elsewhere.*
If Q is an $m\times n$ matrix, with $m<n$, it has a pseudo-inverse $\mathbf{P}$, of dimensions $n\times m$, with $\mathbf{PQ}=\mathbf{I}$.
The inverse of a matrix, if it exists, is unique. Only square matrices have inverses.

DEFINITION 9: *(Symmetric matrices). If* $\mathbf{M}=\mathbf{M}^H$ *then* $\mathbf{M}$ *is* **Hermitian**. *A real Hermitian matrix is* **symmetric** *(that is* $\mathbf{M}=\mathbf{M}^T$*).*
Note that $\mathbf{M}$ must be square to be Hermitian.

THEOREM 3: *The inverse of any symmetric matrix, if it exists, is symmetric. The inverse of a Hermitian matrix, if it exists, is Hermitian.*

Proof: The first follows from the fact that $(\mathbf{A}^T)^{-1}=(\mathbf{A}^{-1})^T$, which is proven in many elementary linear algebra texts (for example [Blyth1986]). The second follows from the fact that $(\mathbf{A}^H)^{-1}=(\mathbf{A}^{-1})^H$ [Smirnov1961]. If $\mathbf{A}$ is Hermitian, then the left hand side is just the inverse of $\mathbf{A}$, so the inverse must be Hermitian. □

5.2.1. Compact Representations of Higher Dimensional Matrices

Vectors are useful as a way of specifying a series of pixel values or image samples. Linear operators represented as matrices are multiplied by the vectors, giving transformed vectors. One such matrix describes the transformation from the set of pixel values at successive frames to the set of intensities at known points in time. Given the matrix which gives the transformation and a vector giving the desired intensities, it is possible to find a vector which when transformed using the matrix gives the closest match to the desired vector. Standard techniques exist for doing this, and are available in numerical packages. When the image is specified in two or more (spatial and/or temporal) dimensions, the vectors of the one dimensional case become matrices, and the matrix defining the transformation is higher dimensional. The Kronecker product is a particularly convenient way of representing these higher dimensional matrices as two dimensional matrices, and the vec operator allows a matrix to be represented as a vector. With these two tools, the higher dimensional problems can be solved using the same library routines as are used for the one dimensional problem.

DEFINITION 10: *(**vec** operator) If* **A** *is an* $n \times m$ *matrix, then* vec **A** *is the column vector which consists of the columns of* **A** *arranged, in order, one above the other.* vec **A** *is necessarily of order* nm.

Example:
If

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

then

$$\text{vec } \mathbf{A} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \\ a_{13} \\ a_{23} \end{bmatrix}$$

DEFINITION 11: *(Kronecker product) If* **A** *is of order* $m \times n$ *and* **B** *is of order* $r \times s$, *then the Kronecker product of the two matrices, denoted* $\mathbf{A} \otimes \mathbf{B}$, *with order* $rm \times ns$ *is defined as*

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}.$$

The next operator is useful for term by term multiplication of two vectors.

DEFINITION 12: *(***diag** *operator) If* **v** *is a vector of order* n *then* diag **v** *is the* $n \times n$ *matrix having the elements of* **v**, *in order* $v_1, v_2, \cdots v_n$ *along the diagonal and zeroes elsewhere.*

To multiply two vectors **a** and **b** term by term, form the diag of one and multiply the resulting matrix by the other: $[\text{diag}(\mathbf{a})\mathbf{b}]_j = \mathbf{a}_j\mathbf{b}_j$.

Many of the properties of Kronecker products are given in Section 2.3 of [Graham1981]; only a few are needed here:

$$(\mathbf{A} \otimes \mathbf{B})^{\mathbf{T}} = \mathbf{A}^{\mathbf{T}} \otimes \mathbf{B}^{\mathbf{T}} \tag{5}$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}) \tag{6}$$

$$\text{vec}(\mathbf{AYB}) = (\mathbf{B}^{\mathbf{T}} \otimes \mathbf{A})\text{vec}\,\mathbf{Y} \tag{7}$$

None of these are defined when the matrices do not conform in such a way as to make the multiplications valid.

5.2.2. Matrix Differentiation

Minimization of functions of matrices and vectors is central to this thesis. To find extrema of a continuous function of a single variable over a continuous domain, it is necessary to differentiate and set the result to zero. When the variable is a matrix, a definition of matrix differentiation is needed.

DEFINITION 13: *(Matrix differentiation) Given a matrix* $\mathbf{A}(t)$, *the derivative of the matrix with respect to a scalar* t *is the matrix whose elements are the derivatives of the corresponding elements in* **A**, *taken with respect to* t.

$$\left[\frac{d}{dt}\mathbf{A}\right]_{ij} = \frac{d}{dt}a_{ij}$$

In a similar fashion, the derivative of a scalar with respect to a column vector is a column vector of corresponding derivatives, and the derivative of a column vector with respect to a scalar is a row vector of corresponding derivatives.

Many properties of vector and matrix differentiation are given in [Graham1981]

5.3. Circulant matrices

Matrices are used to describe transformations. A transformation of interest in this thesis maps frame buffer values to regularly spaced samples of the displayed image. When the sample spacing is the same as the inter-pixel separation, and the image is periodic, the matrix which does this mapping is circulant. A circulant matrix is a special case of a Toeplitz matrix. While the Toeplitz form never appears in the matrices used later in this thesis, it provides the first step in recognizing circulant matrices, and relatively small changes in the formalism produce matrices which are Toeplitz instead of circulant. Pseudo-circulant matrices are rectangular matrices defined analogously to square circulant matrices. They result from sampling the image with a periodicity different from that of the pixels.

5.3.1. Existing Definitions and Results

DEFINITION 14: *(Toeplitz matrices) If* **M** *is an* $n \times n$ *matrix, and* $m_{ij} = m_{i-1,j-1}$ *for* $i, j > 1$, *then* **M** *is said to be* **Toeplitz**.

A Toeplitz matrix is entirely specified by its first row and first column. Each successive row is given by a right shift of the previous row by one element, with a new element introduced in the first column.

DEFINITION 15: *(Circulant matrices). If* **M** *is a Toeplitz matrix, and* $m_{i1} = m_{i-1,n}$ *for* $i > 1$, *then* **M** *is said to be* **circulant**. *For any row vector* **v**, *there exists a circulant matrix with* **v** *as its first row.* **v** *is termed the* **defining vector** *for this matrix.*

Note that a circulant matrix is entirely specified by giving the first row. Each successive row is given by a right rotation of the previous row by one element.

Example:

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ a_n & a_1 & a_2 & \cdots & a_{n-1} \\ a_{n-1} & a_n & a_1 & \cdots & a_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_2 & a_3 & a_4 & \cdots & a_1 \end{bmatrix}.$$

Because of the special properties of circulant matrices, it is as efficient to invert the matrix and obtain the solution by multiplying $\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}$, as it is to solve $\mathbf{Cx} = \mathbf{b}$ directly.

The following theorem gives the inversion procedure.

THEOREM 4: *The inverse of a circulant matrix A, if it exists, is circulant, and it may be computed by the following algorithm:*

1) Compute a vector **f**, which is the inverse discrete Fourier transform, of the first row of **A**.
2) Compute **r**, a vector in which $r_i = 1/f_i$.
3) The first row of $\mathbf{A}^{-1}$ is the inverse discrete Fourier transform of **r**.
4) Other rows are given by rotations of the first row, by the circulant property.

Proof: See [Davis1979], pp. 66-75. □

This method is not only fast, but more stable than matrix inversion with general matrices.

If the order, n, of the matrix is a power of two, or at least highly composite, then the inverse discrete Fourier transform may be computed using the Fast Fourier Transform algorithm in time $O(n \log n)$.

COROLLARY: *A real, symmetric, circulant matrix has a real, symmetric, circulant inverse. A Hermitian, circulant matrix has a Hermitian, circulant inverse.*

In computing the inverse of a symmetric circulant matrix, half of the computation may be avoided because the transforms have no imaginary part.

5.3.2. Circulant Matrices as Sampled Functions

Circulant matrices arise from sampling a continuous, periodic function with each row shifted in phase from the previous one. The phase shift is equal to the period of the function divided by the width of the matrix. Such a function describes the effect of the pixel basis on a periodic image produced by a set of such pixels. If $B(t)$ is the basis, then

$$I(t) = \sum_{q=-\infty}^{\infty} a_q B(t - q\tau)$$

is the intensity in an image formed from pixels with intensity values a_q, and separation τ. If $a_q = a_{q+n}$, then the image is periodic with period $n\tau$, and it may be written as

$$\sum_{l=1}^{n} \sum_{p=-\infty}^{\infty} a_l B((pn - l)\tau + t) \quad .$$

Sampling this at points spaced $j\tau$ apart,

$$I(j\tau) = \sum_l a_l b_{jl}$$

with

$$b_{jl} = \sum_{p=-\infty}^{\infty} B((p\,n - (l-j))\,\tau). \tag{8}$$

The sum in (8) gives the contribution at the point $l\tau$ of a pixel located at $j\tau$, and all replications of that pixel in a periodic image. We define a function

$$\mathcal{B}(t) = \sum_{p=-\infty}^{\infty} B(p\,n\,\tau - t)$$

so that $\mathcal{B}((j-l)\tau)$ is b_{jl}. This function will be used in the generalization of circulant matrices to pseudo-circulant matrices. $\mathcal{B}$ is periodic, with period $n\tau$. This follows because each element of the sum is a copy of the same function translated $n\tau$ from the previous one. In an image with period $n\tau$, the weight a_j affects every nth pixel.

THEOREM 5: *The $n \times n$ matrix* **B** *with elements given by (8) is circulant.*

Proof:

1) **B** is Toeplitz:

$$\begin{aligned} b_{i+1,j+1} &= \mathcal{B}((i+1-(j+1))\tau) \\ &= \mathcal{B}((i-j)\tau) \\ &= b_{ij} \quad . \end{aligned}$$

2) $b_{i1} = b_{i-1,n}$.

$$\begin{aligned} b_{i1} &= \mathcal{B}((i-1)\tau) \\ &= \mathcal{B}(((i-1)-n)\tau) \\ &= b_{i-1,n} \quad . \end{aligned}$$

□

5.3.3. Pseudo-Circulant Matrices

In some cases the sampling in i is not at the same frequency as that in j, for example when an input image is specified at a different sampling rate than the raster. Here $b_{ij} = \sum_{p=-\infty}^{\infty} B(j\tau' - (pn+i)\tau)$, which motivates the following new definition.

DEFINITION 16: *(Pseudo-circulant matrices). If* **B** *is an* $m \times n$ *matrix,* $m \leq n$ *with* $b_{ij} = \mathcal{B}(j\tau' - i\tau) = \sum_{p=-\infty}^{\infty} B(j\tau' - (pn+i)\tau)$. *and* $m\tau = n\tau' = T$, *then* **B** *is called* **pseudo-circulant.** *A square pseudo-circulant matrix is circulant.*

Each row and column of a pseudo-circulant matrix **B** is formed as a fractional rotation of its predecessor, in a sense that should be obvious from the definition. The (constant) fractional rotation taking one row (column) to the next, if applied to the last row (column) yields the first row (column). $\mathcal{B}$ is periodic with period T. Each row consists of n samples from a function with period T. The samples are separated by $\tau' = T/n$ so a row samples exactly one period of the function. Similarly, each column samples exactly one period.

Pseudo-circulant systems can be solved as efficiently as circulant ones, particularly when matrices that are the product of a matrix with its transpose occur.

THEOREM 6: *In the limit of wide matrices, and hence high sampling rates of the function* $\mathcal{B}$, *the product* $\mathbf{BB}^{\mathrm{T}}$ *of a real pseudo-circulant matrix* **B** *with its transpose is symmetric and circulant. If* **B** *is complex, then the product* $\mathbf{BB}^{\mathrm{H}}$ *is Hermitian and circulant, in the same limit.*

Proof: It is straightforward to show that the product of any real matrix with its transpose is symmetric, and that the product $\mathbf{BB}^{\mathrm{H}}$ of any complex matrix with its Hermitian transpose is Hermitian. If **B** is square, then it is sufficient to show that the product $\mathbf{BB}^{\mathrm{T}}$ of a circulant matrix with its transpose is circulant. Consider the first row of the product. Each element is the dot product of the first row of **B** with a rotation of the first row. The first element is the dot product with a rotation by 0, the second with a right rotation by 1, and the last involves a right rotation by -1 (a left rotation by 1). The second row of the product is made up of dot products of the second row with successive right rotations of the second row, by $-1,0,1\ldots$. Throughout the product matrix, the value in an element is determined by the difference in the rotations of the two rows whose dot product is used to form the element. Since the amount of rotation is always shifted in a circulant manner, the product matrix is circulant.

Since the transpose of a circulant matrix is circulant, it follows that $\mathbf{BB}^T$ is also circulant if $\mathbf{B}$ is square.

When the matrix is not square, circulancy of the product only occurs in the limit of large n (wide matrices - it does not depend on the height of the pseudo-circulant, which is the dimension of the product). Each element of the product is given by

$$c_{ij} = \sum_{l=1}^{n} b_{il} b_{lj}^T = \sum_{l=1}^{n} b_{il} b_{jl} = \sum_{l=1}^{n} \beta(l\tau' - i\tau)\beta(l\tau' - j\tau) \quad .$$

$\beta(t)$ is by definition periodic, with period $T = m\tau = n\tau'$. In the limit, as n becomes large, τ' becomes small, since the period, T, remains constant. The normalization condition for the pixel basis function β (from Equation (3)) causes β to become smaller as τ does. In the limit as τ becomes small, this causes the sum to approach the definite integral

$$C(i,j) = \int_0^T \beta(t - i\tau)\beta(t - j\tau)dt \quad .$$

The integral is best visualized by considering an arbitrary periodic function, multiplying it by a copy of itself shifted by a fraction of a period, and integrating the result over a single period. Since β is periodic, with period T, the value of the integrand at any point depends only on the phase shift $(i-j)\tau$ and is periodic in t with period T. Thus the integral depends only on $(i-j)$, and not on either one of i or j. □

For the cases examined in this thesis, the product $\mathbf{BB}^T$ is close to being circulant for relatively small matrices, in the sense that substituting a circulant matrix defined by the first row of the product and then solving yields a solution which is similar to the one which is obtained by finding the solution for the product matrix. For purposes of the algorithms in this thesis, *nearly circulant* is defined as follows:

DEFINITION 17: *If* $\mathbf{A}$ *is a square matrix, and* $\mathbf{A}_1$ *the circulant matrix with the first row of* $\mathbf{A}$ *as its defining vector, then* $\mathbf{A}$ *is* **nearly circulant** *if* $\|\text{vec}(\mathbf{A}_1^{-1}\mathbf{A} - \mathbf{I})\| < \epsilon$, *where* ϵ *is the precision appropriate for the computation.*

The definition leaves open the question of how small ϵ should be, which depends on the application. In this thesis, a matrix is nearly circulant if using the circulant matrix defined by the first row in the place of the complete matrix yields a result which is not visibly different from the result which would have been obtained using the complete matrix. In order to decide whether a matrix is 'nearly circulant' for the purposes of some algorithm, both matrices may be used in a few test cases, and the results compared. A more practical test, useful whenever the product is to be inverted, is to invert the circulant approximation to the product (which is much faster than inverting

the actual product), and then multiply the product by the inverse. The difference of the result from the unit matrix gives a good indication of how nearly the circulant approximates the product for this purpose. How much they may be allowed to differ depends on the application.

Note that the reverse product $\mathbf{B}^T\mathbf{B}$ with $\mathbf{B}$ pseudo-circulant is not, in general, circulant, or even nearly circulant. Similarly, if the dimensions of $\mathbf{B}$ are reversed, that is $m > n$, then the the product $\mathbf{B}^T\mathbf{B}$ is not nearly circulant, but the reverse product $\mathbf{B}\mathbf{B}^T$ is, for large enough values of n.

Algorithms exist for filtering (convolution) in time proportional to the size of the image and to the width of the non-zero part of the filter function. Such algorithms have the advantage of operating on images of indefinite size, since they need to store only as much of the image as is covered by the filter. By contrast, matrix-vector multiplication requires the entire image vector to be in memory. As shown in the next theorem, multiplication by circulant and pseudo-circulant matrices is equivalent to convolution, so space-efficient techniques may be used.

THEOREM 7: *If a vector* $\mathbf{x}$ *of length* n *is constructed so that* $x_j = f(j\tau)$, *where* f *is periodic, with period* $T = n\tau$, *and a pseudo-circulant matrix* $\mathbf{C}$ *of dimensions* $n \times m$ *is constructed so that*

$$C_{ij} = \sum_{p=-\infty}^{\infty} B(i\tau' - (pn + j)\tau), \quad \tau' = n\tau/m,$$

then the product $\mathbf{y} = \mathbf{C}\mathbf{x}$ *approximates the convolution of* B *with* f *at the points* $t = j\tau$. *The limit of the sum as* $\tau \to 0$ *is the convolution integral.*

Proof: A single element of the product is $y_i = \sum_{j=1}^{n} C_{ij} x_j$. But by the definitions of $\mathbf{C}$ and $\mathbf{x}$,

$$y_i = \sum_{j=1}^{n} \sum_{p=-\infty}^{\infty} B(i\tau' - (pn + j)\tau) f(j\tau)$$

$$= \sum_{l=-\infty}^{\infty} B(i\tau' - l\tau) f(l\tau),$$

since f is periodic. In the limit as $\tau \to 0$, y_i is the convolution of B with f. ⊏

Digital filtering is a simple form of numerically computing the convolution integral of the filter function with the function (image) being filtered. The values of the filter that are used for filtering when the filter is the function $B(t)$ are given by the elements of the matrix.

5.4. Algorithms

This section gives algorithms for finding a solution for the direct method, and for the frequency space method. In each case, the algorithms for solving in one dimension are given first and then they are extended to two dimensions. Solutions are given in terms of finding matrices and solutions to matrix vector problems. Specific matrix elements are found for particular cases in Chapter 8.

5.4.1. Direct Solution

If the pixels in a device are independent, the values of the pixels may be related to the intensities at evenly spaced sample points using a pseudo-circulant matrix. How this is done is shown at the beginning of Section 5.4.

In such a system, the input vector $\mathbf{x}$ gives the pixel values; the output vector $\mathbf{b}$, gives the intensities produced at the sample points. When the input vector is multiplied by the matrix, the output vector is the product. When $\mathbf{x}$ is given, $\mathbf{b}$ may be computed as $\mathbf{Bx}$ and used in a simulation of a device with pixels that are constructed from several pixels of a real device. When $\mathbf{b}$ is given, then the system $\mathbf{Bx}=\mathbf{b}$ may be solved for $\mathbf{x}$, giving the frame buffer values required to produce the intensities specified by $\mathbf{b}$. This is the mathematical basis for all of the minimization techniques that follow.

The pseudo-circulant system $\mathbf{Bx}=\mathbf{b}$, with $\mathbf{B}$ $m \times n$, $(m < n)$, is under-determined. That is, there are more unknowns than equations to satisfy. Thus there is a non-empty vector space, called the null space of $\mathbf{B}$. The sum of a solution vector $\mathbf{x}$ with any vector in the null space is another solution vector. The vector with zero projection onto the null space of $\mathbf{B}$ has the least extra information added to it, and in some sense is the best choice. This vector is obtained by solving $(\mathbf{BB}^{\mathbf{T}})\mathbf{v}=\mathbf{b}$ for $\mathbf{v}$, and then setting $\mathbf{x}=\mathbf{B}^{\mathbf{T}}\mathbf{v}$ [Golub1983]. If $\mathbf{B}$ is wide enough, then $\mathbf{BB}^{\mathbf{T}}$ is nearly circulant, and its inverse is easily computed as outlined in Theorem 4. The result is then $\mathbf{x}=\mathbf{B}^{\mathbf{T}}(\mathbf{BB}^{\mathbf{T}})^{-1}\mathbf{b}$.

The complementary system $\mathbf{Ax}=\mathbf{b}$, with $\mathbf{A}=\mathbf{B}^{\mathbf{T}}$ and $\mathbf{B}$ pseudo-circulant as before, is over-determined. This corresponds to having more sample values than pixels. It is generally impossible to find an exact solution. In this case, the best possible is a vector which minimizes a norm of the residual vector $\mathbf{r}=\mathbf{Ax}-\mathbf{b}$. Minimizing the l_1 norm of the residual keeps the average error as small as possible. Minimizing the l_∞ norm keeps the maximum error as small as possible. The l_2 norm is a compromise between the two.

The problem of minimizing the l_1 or l_∞ norm of a matrix-vector system is a linear programming problem, which can be solved with or without constraints, using available software. To solve the l_2 minimization problem, the product $\mathbf{r}^{\mathbf{H}}\mathbf{r}$ is

minimized. The risk of this method is that in forming products of matrices with their transposes, small elements might be lost. In this case, the small elements are not significant in the result, so it does not matter. Most standard software requires all matrices to be real. When $\mathbf{r}$ is real, the product is $\mathbf{r}^T\mathbf{r}$. For reasons which will become apparent in the next section, we keep $\mathbf{r}$ complex. In any case the product is real.

To minimize $\mathbf{r}^H\mathbf{r} = (\mathbf{Ax}-\mathbf{b})^H(\mathbf{Ax}-\mathbf{b})$, expand the product

$$(\mathbf{Ax}-\mathbf{b})^H(\mathbf{Ax}-\mathbf{b}) = \mathbf{x}^H\mathbf{A}^H\mathbf{Ax} - \mathbf{x}^H\mathbf{A}^H\mathbf{b} - \mathbf{b}^H\mathbf{Ax} + \mathbf{b}^H\mathbf{b} \ .$$

The last term is a known real scalar. The second and third terms are complex conjugates of each other: their sum is a real scalar. Thus the first term is real since all of the other terms are real. It is sufficient to minimize $\mathbf{x}^H(\mathbf{A}^H\mathbf{A})\mathbf{x} - (\mathbf{b}^H\mathbf{Ax} + (\mathbf{b}^H\mathbf{Ax})^H)$. Differentiating with respect to $\mathbf{x}$ [Dwyer1948] and setting the result to zero:

$$(\mathbf{A}^H\mathbf{A} + \mathbf{A}^T\mathbf{A}^*)\mathbf{x} = \mathbf{A}^T\mathbf{b}^* + \mathbf{A}^H\mathbf{b} \tag{9}$$

or $\mathrm{Re}(\mathbf{A}^H\mathbf{A})\mathbf{x} = \mathrm{Re}(\mathbf{A}^H\mathbf{b})$. The l_2 norm solution to the minimization problem is the solution to this new problem:

$$\mathbf{x} = (\mathbf{A}^H\mathbf{A} + (\mathbf{A}^H\mathbf{A})^T)^{-1}(\mathbf{A}^T\mathbf{b}^* + \mathbf{A}^H\mathbf{b}) \ . \tag{10}$$

Since $\mathbf{A}^H$ is pseudo-circulant, $\mathbf{A}^H\mathbf{A}$ is real and nearly circulant, but not necessarily symmetric. The sum of a matrix and its transpose is always symmetric. Hence the inverse may be quickly computed using the algorithm of Theorem 4. If $\mathbf{A}$ and $\mathbf{b}$ are real, the solution is slightly simpler: $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{b})$.

These equations can be expanded to higher dimensions using the Kronecker matrix notation. Let $\mathbf{X}$ be a matrix of pixel values to be determined for a two dimensional (one spatial and one temporal) image. The rows of $\mathbf{X}$ correspond to frames in the image; the columns correspond to the sequence of intensities for particular pixels. If $\mathbf{A}_1^T$ and $\mathbf{A}_2^T$ are the matrices which would be used in the one dimensional cases of space and time, respectively, then $\mathbf{A}_2^T\mathbf{X}$ gives the image which would be displayed by the device, if it had pixels with no spatial spread, and $\mathbf{XA}_1$ gives the image which would be displayed if pixels had no temporal spread. $\mathbf{A}_2^T\mathbf{XA}_1$ gives the image on a device with pixels having temporal and spatial pixels spreads. If $\mathbf{B}$ gives the sample values, then the difference $\mathbf{A}_2^T\mathbf{XA}_1 - \mathbf{B}$ is a matrix giving the errors at samples. The vec of this matrix is a vector containing the same values. It can be

minimized using vector minimization techniques. Since

$$\text{vec}\,(A_2^T X A_1) = (A_1 \otimes A_2)\text{vec}\,X,$$

the equivalent problem is to minimize

$$\|Ax - b\|, \tag{11}$$

where $A = A_1 \otimes A_2$, $x = \text{vec}\,X$, and $b = \text{vec}\,B$, for matrices A_1, A_2, X and B. For the l_2 norm, $(A^H A + A^T A^*)x = A^T b^* + A^H b$ as before. This equation may be solved directly in two ways: by forming the inverse of the matrix expression on the left or by using Gaussian elimination to find x. By using the structure of the matrix, we can reformulate the problem and the computation can be significantly reduced. Expanding

$$((A_1 \otimes A_2)^H (A_1 \otimes A_2) + (A_1 \otimes A_2)^T (A_1 \otimes A_2)^*)\text{vec}\,X$$
$$= (A_1 \otimes A_2)^T \text{vec}\,B^* + (A_1 \otimes A_2)^H \text{vec}\,B \quad ,$$

$$\Rightarrow ((A_1^H \otimes A_2^H)(A_1 \otimes A_2) + (A_1^T \otimes A_2^T)(A_1 \otimes A_2)^*)\text{vec}\,X$$
$$= (A_1^T \otimes A_2^T)\text{vec}\,B^* + (A_1^H \otimes A_2^H)\text{vec}\,B \quad ,$$

(by (5))

$$\Rightarrow ((A_1^H A_1) \otimes (A_2^H A_2) + (A_1^T A_1^*) \otimes (A_2^T A_2^*))\text{vec}\,X$$
$$= (A_1^T \otimes A_2^T)\text{vec}\,B^* + (A_1^H \otimes A_2^H)\text{vec}\,B \quad ,$$

(by (6))

$$\Rightarrow (((A_1^H A_1) + (A_1^T A_1^*)) \otimes ((A_2^H A_2) + (A_2^T A_2^*)))\text{vec}\,X$$
$$= (A_1^T \otimes A_2^T)\text{vec}\,B^* + (A_1^H \otimes A_2^H)\text{vec}\,B \quad ,$$

$$\Rightarrow \text{vec}\,(((A_2^H A_2) + (A_2^T A_2^*))^T X ((A_1^H A_1) + (A_1^T A_1^*)))$$
$$= \text{vec}\,(A_2 B^* A_1^T) + \text{vec}\,(A_2^* B A_1^H) \quad ,$$

(by (7))

$$\Rightarrow ((A_2^H A_2) + (A_2^T A_2^*))^T X ((A_1^H A_1) + (A_1^T A_1^*)) = A_2 B^* A_1^T + A_2^* B A_1^H \quad .$$

Assuming that the appropriate inverses exist,

$$X = ((A_2^H A_2) + (A_2^T A_2^*))^{-T} A_2 B^* A_1^T + A_2^* B A_1^H ((A_1^H A_1) + (A_1^T A_1^*))^{-1} \quad . \tag{12}$$

Now both terms of $(\mathbf{A}_2^{\mathrm{H}}\mathbf{A}_2)+(\mathbf{A}_2^{\mathrm{T}}\mathbf{A}_2^{*})$ are Hermitian matrices, and they are complex conjugates of each other, so the sum is real and symmetric. This means that the inverse-transpose appearing in (12) is the same as the inverse, so

$$\mathbf{X}=((\mathbf{A}_2^{\mathrm{H}}\mathbf{A}_2)+(\mathbf{A}_2^{\mathrm{T}}\mathbf{A}_2^{*}))^{-1}\mathbf{A}_2\mathbf{B}^{*}\mathbf{A}_1^{\mathrm{T}}+\mathbf{A}_2^{*}\mathbf{B}\mathbf{A}_1^{\mathrm{H}}((\mathbf{A}_1^{\mathrm{H}}\mathbf{A}_1)+(\mathbf{A}_1^{\mathrm{T}}\mathbf{A}_1^{*}))^{-1} \ .$$

Each of the inverses is of a small matrix. The order of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ is the product of the orders of $\mathbf{A}_1^{\mathrm{H}}\mathbf{A}_1$ and $\mathbf{A}_2^{\mathrm{H}}\mathbf{A}_2$. As a result this is a much more efficient technique of solution than treating the entire system as a whole. If $\mathbf{A}_1^{\mathrm{H}}\mathbf{A}_1$ and $\mathbf{A}_2^{\mathrm{H}}\mathbf{A}_2$ are symmetric circulant, then each of the inverses can be found in time $n_i \log n_i$ time, where n_i is the order of $\mathbf{A}_i^{\mathrm{H}}\mathbf{A}_i$.

5.4.2. Frequency Space Minimization

The methods above are useful for the direct minimization approach outlined in the last chapter. Circulant matrices arise in the frequency space minimization as well, but for different reasons. In this method, the Fourier expansions of the desired image and an (as yet unknown) image formed from device pixels are compared, with the difference between the two being minimized. A more sophisticated variant on this weights each Fourier coefficient according to the sensitivity of the human visual system at the corresponding frequency. In one dimension, the problem is to find the set of amplitudes a_p which when multiplied by copies of a function representing the pixel shape, $B(\xi)$, give

$$V_n=\mathcal{F}[I(\xi),\varsigma_n]-\mathcal{F}\left[I'(\xi),\varsigma_n\right]<T(\varsigma_n) \ , \tag{13}$$

where $T(\varsigma_n)$ is the contrast sensitivity at frequency ς_n, $I(\xi)$ is the desired image, and $I'(\xi)=\sum_{p=-\infty}^{\infty} a_p B(\xi-p\delta)$ is the image produced on the display. In practice $I(\xi)$ is taken as being of finite duration or periodic, so that the Fourier series can be used in the place of the transform. Since any set of weights a_p which satisfies (13) is equally good, the approach is to minimize

$$\left|\left|\frac{1}{T(\varsigma_n)}\left(\mathcal{F}[I(\xi),\varsigma_n]-\mathcal{F}[I'(\xi),\varsigma_n]\right)\right|\right| \ .$$

Now let the coefficients in the complex Fourier series of I and I' be z_n and z_n' respectively. The objective is to minimize $\|\mathrm{diag}\ (\mathbf{W})(\mathbf{z}'-\mathbf{z})\|$, where $W_n=1/T(\varsigma_n)$. From Theorem 1,

$$z_n' = \frac{2\sqrt{2\pi}}{T}\sum_{j=1}^{N} a_j \mathcal{F}[B(\xi - j\delta), \varsigma_n] = \frac{1}{W_n}\sum_{j=1}^{N} b_{nj} a_j \quad ,$$

with $b_{nj} \equiv \frac{2\sqrt{2\pi}W_n}{T}\mathcal{F}[B(\xi - j\delta), \varsigma_n]$. (Note that n is not the width of the matrix in this case.) If $y_n = W_n z_n$, we are minimizing the residual of a matrix vector system: $\|\mathbf{Ba} - \mathbf{y}\|_2$, where $\mathbf{B}$ is not necessarily pseudo-circulant.

If there are no constraints on the values of $\mathbf{a}$, there is an efficient way of computing the l_2 norm solution. The solution is $\mathbf{a} = (\mathbf{B}^H\mathbf{B} + \mathbf{B}^T\mathbf{B}^*)^{-1}(\mathbf{B}^T\mathbf{y}^* + \mathbf{B}^H\mathbf{y})$, as shown in Section 5.4, Equation (10). This solution can be written as $\mathbf{a} = \mathbf{C}^{-1}\mathbf{x}$, where $\mathbf{C} = (\mathbf{B}^H\mathbf{B} + \mathbf{B}^T\mathbf{B}^*)$, and $\mathbf{x} = (\mathbf{B}^T\mathbf{y}^* + \mathbf{B}^H\mathbf{y})$. Expanding,

$$c_{jl} = \frac{2\sqrt{2\pi}}{T}\sum_n W_n^2(\mathcal{F}[B(\xi - j\delta), \varsigma_n]^*\mathcal{F}[B(\xi - l\delta), \varsigma_n] + \mathcal{F}[B(\xi - l\delta), \varsigma_n]^*\mathcal{F}[B(\xi - j\delta), \varsigma_n])$$

and $x_l = \sum_n W_n^2(\mathcal{F}[B(\xi - l\delta), \varsigma_n]^* z_n + z_n^*\mathcal{F}[B(\xi - l\delta), \varsigma_n])$. Using the shift theorem for Fourier transforms (Theorem A.3), c_{jl} may be re-written as follows:

$$c_{jl} = \frac{2\sqrt{2\pi}}{T}\sum_n W_n^2(e^{i\varsigma_n(j-l)\delta} + e^{i\varsigma_n(l-j)\delta})\mathcal{F}[B(\xi), \varsigma_n]^*\mathcal{F}[B(\xi), \varsigma_n]$$

$$= \frac{2\sqrt{2\pi}}{T}\sum_n W_n^2\cos(\varsigma_n(j-l)\delta)\mathcal{F}[B(\xi), \varsigma_n]^*\mathcal{F}[B(\xi), \varsigma_n] \quad .$$

Even though $\mathbf{B}$ is neither pseudo-circulant nor real, $\mathbf{C}$ is circulant and symmetric, independent of the matrix size. The proof follows:

1) $\mathbf{C}$ is symmetric Toeplitz, that is, c_{jl} is real and depends on $|j - l|$, but not on $j + l$. The product of any complex number with its complex conjugate is real, so each term in the sum is real.

2) $c_{j1} = c_{j-1,N}$, $j > 1$. It is sufficient that corresponding terms in the series represented by the left and right hand sides of the equation agree. That is,

$$W_n^2\cos(\varsigma_n(j-1)\delta)\mathcal{F}[B(\xi), \varsigma_n]^*\mathcal{F}[B(\xi), \varsigma_n] \equiv W_n^2\cos(\varsigma_n(j-1-N)\delta)\mathcal{F}[B(\xi), \varsigma_n]^*\mathcal{F}[B(\xi), \varsigma_n],$$

and since $\varsigma_n = 2n\pi/N\delta$, the identity is true because the cosine is periodic.

Taken together, 1) and 2) prove that $\mathbf{C}$ is circulant. □

The two dimensional problem is a straightforward extension of this one-dimensional analysis, as long as the two dimensional basis can be expressed as the product of two one dimensional basis functions. In such a case the basis function is called *separable*. Let $B_t(t)$ and $B_x(x)$ be the temporal and (one dimensional) spatial

pixel intensity functions, and $I(x,t)$ be the desired image. If W_{rs} is the two dimensional analogue to W_n with r indexing spatial frequency and s indexing temporal frequency, then the problem is to find the set of amplitudes a_{pq} which minimizes

$$\sum_{r,s} W_{rs}^2 \left(\mathcal{S}[I(x,t),k_r,\omega_s] - \mathcal{S}[I'(x,t),k_r,\omega_s] \right)^2 ,$$

with $I'(x,t) = \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} a_{pq} B_x(x-q\kappa) B_t(t-p\tau)$, or, more compactly, minimizes $\sum_{r,s} \|\mathbf{Z}'_{rs} - \mathbf{Z}_{rs}\|^2 = \|\text{vec } \mathbf{Z}' - \text{vec } \mathbf{Z}\|_2$, with $Z_{rs} = W_{rs}\, \mathcal{S}[I(x,t),k_r,\omega_s]$

and $Z'_{rs} = W_{rs}\, \mathcal{S}[\sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} a_{pq} B_x(x-q\kappa) B_t(t-p\tau), k_r, \omega_s]^2$.

Here $\mathcal{S}[I(x,t),k_r,\omega_s]$ is the (r,s) element of the Fourier series of $I(x,t)$ (see the Appendix). From Theorem 2, in Section 5.1,

$$\mathcal{S}[\sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} a_{pq} B_x(x-q\kappa) B_t(t-p\tau), k_r, \omega_s]^2$$

$$= \sum_{i,j} \frac{2\pi}{4DT} a_{ij}\, \mathcal{F}[B_t(t-j\tau), -\omega_s]\, \mathcal{F}[B_x(x-i\kappa), -k_r]$$

$$= \sum_{i,j} b_{ir}^{(x)} a_{ij} b_{js}^{(t)} .$$

Thus $Z'_{rs} = W_{rs} \sum_{i,j} b_{ir}^{(x)} a_{ij} b_{js}^{(t)}$,

or $\mathbf{z}' = \text{vec } \mathbf{Z}' = \text{diag (vec } \mathbf{W}) \text{vec } (\mathbf{B}^{(x)T} \mathbf{A} \mathbf{B}^{(t)})$

$$= \text{diag (vec } \mathbf{W}) (\mathbf{B}^{(t)T} \otimes \mathbf{B}^{(x)T}) \text{vec } \mathbf{A} .$$

If $\mathbf{W}$ is separable into $W^{(x)} W^{(t)}$, the expression for Z' is simpler:

$$\mathbf{Z}' = (\mathbf{W}^{(x)T} \mathbf{B}^{(x)T}) \mathbf{A} (\mathbf{B}^{(t)} \mathbf{W}^{(t)}) .$$

In the non-separable case, the minimization is $\|\mathbf{B}'\mathbf{a} - \mathbf{z}\|_2$, with $\mathbf{B}' = \text{diag (vec } \mathbf{W})(\mathbf{B}^{(t)T} \otimes \mathbf{B}^{(x)T})$, and $\mathbf{a} = \text{vec } \mathbf{A}$. The minimization analogous to the one-dimensional minimization above (equation (9)), except that the inverse cannot be obtained cheaply by taking advantage of circulancy. Thus a more general matrix vector solution technique must be used. From Equation (9).

$$\mathrm{Re}(\mathbf{B'}^{H}\mathbf{B'})\mathbf{a} = \mathrm{Re}(\mathbf{B'}^{H}\mathbf{z}) \ ,$$

$$\mathbf{B'}^{H}\mathbf{B'} = \mathrm{diag}\ (\mathrm{vec}\ \mathbf{W})^{T}(\mathbf{B}^{(t)*}\otimes\mathbf{B}^{(x)*})(\mathbf{B}^{(t)T}\otimes\mathbf{B}^{(x)T})\mathrm{diag}\ (\mathrm{vec}\ \mathbf{W})$$

$$= \mathrm{diag}\ (\mathrm{vec}\ \mathbf{W})(\mathbf{B}^{(t)*}\mathbf{B}^{(t)T}\otimes\mathbf{B}^{(x)*}\mathbf{B}^{(x)T})\mathrm{diag}\ (\mathrm{vec}\ \mathbf{W}) \ .$$

The central product of this expression is the Kronecker product of the two real matrices used in the one-dimensional formulation. W is real. From this it is clear that the entire product is real.

If W is separable, vec $\mathbf{z'} = (\mathbf{B}^{(t)}\mathbf{W}^{(t)})^{T}\otimes(\mathbf{W}^{(x)T}\mathbf{B}^{(x)T})$vec A and the quantity to be minimized is $\|(\mathbf{B}_1\otimes\mathbf{B}_2)\mathrm{vec}\ \mathbf{A} - \mathrm{vec}\ \mathbf{Z}\|$, which has the same form as (11). Thus the solution is

$$\mathbf{A} = ((\mathbf{B}_2^{H}\mathbf{B}_2) + (\mathbf{B}_2^{T}\mathbf{B}_2^{*}))^{-T}\mathbf{B}_2\mathbf{Z}^{*}\mathbf{B}_1^{T} + \mathbf{B}_2^{*}\mathbf{Z}\mathbf{B}_1^{H}((\mathbf{B}_1^{H}\mathbf{B}_1) + (\mathbf{B}_1^{T}\mathbf{B}_1^{*}))^{-1} \ ,$$

from (12). Referring back to the one dimensional frequency space method at the start of this section, the matrices to be inverted are the same as the ones which are inverted in the one dimensional spatial and temporal cases, for the same basis and image width. Thus they are symmetric and circulant.

5.5. Summary

Most of the definitions in this chapter are not original; the definitions of pseudo-circulant and nearly circulant matrices are. Notations for matrices, their complex conjugates, regular and Hermitian transpose vary from source to source. Some definitions are relatively obscure, being either too new or considered of too little centrality to mathematics to be well known. The vec operator, Kronecker product and matrix differentiation are relatively new; Toeplitz matrices, and circulant matrices in particular are not generally considered central enough to be introduced at the undergraduate level. Several definitions only appear for completeness: the vector norm, matrix inverse and symmetric matrices fit in this category.

The theorem relating an infinite sum of weighted basis functions to its Fourier series is new, as is its two dimensional extension. Theorems 5-7, dealing with ways in which circulant and pseudo-circulant matrices arise are all new.

The algorithm for minimizing the l_2 norm of a vector expression is well known; it is usually presented for real matrices, and some care is required in extending it to complex matrices. The concise presentation using matrix differentiation for minimizing the l_2 norm appears, among other places, in Rau's statistics text [Rau1965]. Dierckx used the Kronecker product notation to extend this presentation to two dimensions with B-spline basis functions serving the same purpose as the pixel basis functions used here [Dierckx1977]. The extension to complex matrices does not appear in either of these references.

Minimization Approaches

When the problem was introduced formally in Chapter 4, several alternative criteria for measuring the difference between images were suggested: matching at sample points, matching averages between sample points, and matching in frequency space. Although other criteria are possible, these three are representative of ways of solving the problem, however it is posed.

Regardless of the method of matching, an exact solution may be impossible: intensity values greater than the device maximum or less than the device minimum can be required to match the images. There are two solutions to this problem. One puts constraints on the input, only allowing images that do not cause problems. The constraints typically are contrast restrictions. The other uses constrained minimization. Each approach has advantages. The main advantage of using constrained minimization is that any input image may be supplied and a valid solution is returned. In many images, the locations of changes in intensity are more important than the absolute intensity; for such images a reduction in contrast is better than using constraints, since edges may be lost in regions where constraints are active. There may also be tradeoffs between contrast and accuracy which depend on unknown characteristics of the visual system. Unconstrained minimization also runs more quickly, and can be implemented as filtering.

Whether the system is constrained or not, and regardless of what is being minimized, there remains the choice of norm used to measure the distance of a given solution from the ideal image. Here only the l_1, l_2, and l_∞ norms are considered, since techniques for minimizing them are relatively well known, and they cover the spectrum of norms well enough for some comparisons to be made.

The previous chapter provided the mathematical foundations for all three methods of solution. In Chapter 7, methods of solution using the tools already presented are outlined. In the Chapter 8, two profiles are used as examples to illustrate the use of the techniques outlined in general in this chapter.

This chapter begins with the three methods: it considers first matching images at sample points, then matching integrals between sample points, and lastly frequency space minimization. After this the two methods of avoiding values which lie outside the range of the device are considered; the first constraining the input, the second constraining the output. Finally using filtering to obtain results equivalent to the unconstrained minimization is discussed.

6.1. Minimizing Differences at Sample Points

Just as the basis functions are assumed to be equally spaced, so the sampling density of the ideal function is assumed uniform. This could be relaxed, at the cost of some efficiency. The spacing of the basis functions and the spacing of samples need not be the same. If the density of basis functions is greater than the density of sample values, there are multiple solutions, and a strategy must be adopted to select among them. If the density of basis functions is less than the density of sample values, only an approximate solution is possible. An approximate minimization technique must be used to find the best solution. If the two densities are equal then an exact solution may be possible, depending on the contrast of the image.

In one dimension, matching sampling values is formalized as follows. We are given a sampling of an ideal image $I(t)$, for sample points $t_k = k\tau$, a basis function $B(t)$, and pixel separation τ'. We want to find the set of values a_j which, when multiplied by $B(t_j)$ and summed, gives a function $J(t)$, with $J(k\tau) \equiv I(k\tau)$. In the case $\tau = \tau'$, some simplifications exist. When $I(t)$ is periodic, the matrix which is used in the solution is circulant. This does not mean that the image needs to be periodic. When it is not periodic, then the image can be extended at both ends with enough blank space that pixels used to display the end of one period have no significant residual effect on the displayed image at the start of the next period. Mathematically,

$$I(k\tau) = \sum_{p=-\infty}^{\infty} a_p B(k\tau - p\tau) \quad .$$

If $I(t)$ is periodic, with period $N\tau = T$, $a_k \equiv a_{k-Ni}$, for all integer values of i. Then it is possible to write

$$I(k\tau) = \sum_{q=-\infty}^{\infty} \sum_{j=1}^{N} a_j B((k-(j+Ni))\tau) \quad ,$$

and re-arrange the order of summation to

$$I(k\tau) = \sum_{j=1}^{N} a_j \sum_{q=-\infty}^{\infty} B((k-(j+Nq))\tau) \quad .$$

If we let $b_{kj} = \sum_{q=-\infty}^{\infty} B((k-(j+Nq))\tau)$, the summation becomes $I(k\tau) = \sum_{j=1}^{N} b_{kj} a_j$, or $\mathbf{x} = \mathbf{B}\mathbf{a}$. Since sample spacing is the same as the separation of the basis functions, $\mathbf{a}$ has the same number of elements as $\mathbf{x}$, meaning that the matrix $\mathbf{B}$ is square. It has already been established in Section 5.3 that any $\mathbf{B}$ formed in this way is circulant.

The more general case of different (but uniform) sample spacings is

$$I(k\tau) = \sum_{p=-\infty}^{\infty} a_p B(k\tau - p\tau') \quad ,$$

with $N\tau = M\tau' \equiv T$, so that $a_k \equiv a_{k-Nl}$ as before. Note that the correspondence of $N\tau = M\tau'$ can always be arranged, possibly by extending the blank ends of the desired image. Then

$$I(k\tau) = \sum_{j=1}^{M} a_j \sum_{q=-\infty}^{\infty} B(k\tau - (j+Mq)\tau') \quad ,$$

or $\mathbf{x} = \mathbf{B}\mathbf{a}$, with $b_{kj} = \sum_{q=-\infty}^{\infty} B(k\tau - (j+Nq)\tau')$.

In this case $\mathbf{B}$ is not square, but if $N > M$, it is pseudo-circulant. It is for this reason that the results of Section 5.3.3 were introduced.

For some basis functions the infinite sum may be found in closed form. In those cases in which it cannot, the sum should be calculated from the smallest appreciable value to largest in order to minimize numerical error (as long as the non-appreciable values are known to have a non-appreciable sum – which is the case for all reasonable pixel bases). Pixel basis functions are generally monotonic on each side of the origin, so the smallest to largest ordering is toward the origin.

6.2. Minimizing Differences of Averages

One of the problems of sampling is that it is not necessarily average preserving. It is a desirable property of any procedure applied to an image that the average intensity of the entire image and any region within the image be the same in the ideal specification as it is in a realization on the device. The average of a function $f(t)$ over an interval $t_1..t_2$ is

$$\frac{\int_{t_1}^{t_2} f(t)dt}{t_2 - t_1} .$$

Minimizing the difference between averages over an interval is equivalent, then, to minimizing the difference between integrals over the same interval. For the moment, consider only the case of equal spacing ($\tau = \tau'$). Mathematically, matching integrals is finding the set of weights a_j which give

$$V_k = \int_{k\tau}^{(k+1)\tau} I(t)dt = \int_{k\tau}^{(k+1)\tau} \sum_{j=-\infty}^{\infty} a_j B(t - j\tau)dt = \int_{k\tau}^{(k+1)\tau} \sum_{j=-\infty}^{\infty} \sum_{l=0}^{N} a_l B(t - (jN + l)\tau)dt$$

for every interval k, or minimizing the error in the approximation. Now, it is possible to transform this into an instance of the earlier problem as follows:

$$V_k = \int_{k\tau}^{(k+1)\tau} \sum_{j=-\infty}^{\infty} \sum_{l=0}^{N} a_l B(t - (jN + l)\tau)dt = \sum_{j=-\infty}^{\infty} \sum_{l=0}^{N} a_l \int_{k\tau}^{(k+1)\tau} B(t - (jN + l)\tau)dt .$$

Let $u = t - (jN + l)\tau$ and $\mathcal{B}(t,\tau) = \int_{t}^{t+\tau} B(u)du$.

Then $$\int_{k\tau}^{(k+1)\tau} B(t - (jN + l)\tau)dt = \int_{(k-jN-l)\tau}^{(k-jN-l+1)\tau} B(u)du = \mathcal{B}((k - jN - l)\tau, \tau) .$$

So $V_k = \sum_l a_l \sum_j \mathcal{B}((k - jN - l)\tau, \tau)$, which has the same form as for value matching. The new basis function $\mathcal{B}$ depends on τ, which is assumed constant. Thus, matching integrals is the same problem as matching sample values, but with a different choice of basis function. This is easily extended to the case of $\tau \neq \tau'$.

It is possible to extend this to averages over multiple intervals, and then further, to include weighted averages with weighting functions which have finite integrated values. To extend it to multiple intervals, the new basis function is formed from the integral over multiple intervals, rather than one interval. With weighted averages, the

new basis function is formed from the weighted integral. Using weighted averages over multiple intervals allows a form of filtering of the image, and is common in antialiasing.

6.3. Frequency Space Minimization

The general method of doing frequency space minimization was discussed in the Section 5.4.2. Finding an unconstrained solution in the l_2 norm with a separable approximation for the human visual system's sensitivity function requires the inversion of circulant matrices, or the Kronecker product of circulant matrices, so that the solution process is computationally efficient.

Various approximations can be used for the sensitivity function giving the relative importance of the different frequencies in the sum. Several of them are discussed in the last part of Chapter 3. The simplest approximation is to use an infinite series, assuming all frequencies are equally important. In most cases the sum is not available in closed form, so it is cheaper to use an approximation in which all frequencies are weighted equally up to the cutoff of the human visual system. Because of the shape of the human visual system response function, this is a better approximation than using the infinite sum, and is equivalent to using Watson's window of visibility approach in one dimension (see Chapter 3). Better approximations use weights proportional to the visibility of the corresponding frequencies, based on the spatial or temporal response functions.

One approximation is Watson's window of visibility, the product of a spatial and a temporal response function which are both constant up to a finite cutoff value. A better approximation is a product of good approximations of the temporal and spatial response functions, at intermediate values of the spatial and temporal frequency domains respectively, while the best approximation is close to the actual (non-separable) spatiotemporal response function, such as the approximation used to draw Figure 7 in Chapter 3.

In both the direct and frequency space minimization, the solution lies in solving a matrix vector problem. The algorithm for frequency space minimization was given in the last chapter. In the unconstrained l_2 case, it involved solving a circulant system. If there are constraints, then it is still a matrix-vector problem (with a complex matrix), although not as simple to solve as inverting the matrix. This is also the case for the other norms. In the direct minimization case, a circulant or pseudo-circulant matrix-vector problem is solved. If it is circulant, then the choice of norm does not affect the solution unless there are also constraints: the solution is exact. For each choice, the matrices involved have been described in general terms in Chapter 5 (i.e. for a general pixel profile). The elements of specific matrices are given in Chapter 7.

6.4. Domain Restrictions – Limiting Contrast

Whatever matching technique is used, the calculated intensities are displayed on a device having a bounded intensity range. The bounds on the real device can be taken as 0 and 1 without loss of generality. Less realistically the device is assumed capable of producing a continuous range of pixel intensities. The best unconstrained approximations to some images have all calculated pixel intensities within the intensity range of the device. These images may be displayed as is. As shown below the important image characteristic is the contrast. For some pixel basis functions it is possible to calculate the maximum contrast a representable image may have. In all others a bound is at least possible. In this section it is shown how to find the maximum or a bound on it.

As is mentioned both in the introduction and two chapters later where the problem is defined, unconstrained minimization is always possible if the input contrast is adjusted appropriately. If constrained minimization is not used, two approaches are available. The first is to provide the contrast requested, as long as the requested contrast is not greater than the maximum available; when the contrast of the input is too great, the result is undefined. (This is like constrained minimization, in an instance in which the constraints are not needed). The second is to map a known range of the input (such as the 0..1 range) to the available input contrast range in a linear fashion. Both methods are equivalent in the sense that if the mapping used by the second method is known, its inverse can be manually applied to the input (effectively undoing it) to provide the effect of the first method. Similarly the mapping can be applied before using the first method, giving the same effect as the second method. The first method is more convenient when, for whatever reason, contrast must be specified in absolute terms; the second is more convenient when device independence is an asset.

Whichever approach is taken, a measure of the allowable contrast is required. For unconstrained minimization with the l_2 norm, the problem has been reduced to one of solving a matrix vector system, which can be represented as a matrix-vector multiplication (with the inverse matrix and the right-hand side vector), regardless of whether the method is direct or in frequency space. Because the l_2 norm is the most appropriate for problems of this type (as shown in Section 8.1.4), it is the only norm which is considered at here. There are two possible ways of ensuring that the output image is within the limits of the device. The more appealing is to derive a value for the allowable contrast, so that any image with contrast within the limits of the allowable contrast, when multiplied by the inverse matrix, yields an output in the 0..1 range. This method gives the user the value of the allowable contrast and the choice of transforming the input to fit in the allowable contrast or paying the penalty of using a constrained minimization package. A simpler, but less appealing method is to find

the minimum and maximum values at the output for an input vector which is in the 0..1 range. An appropriate affine transformation may then be applied to any output vector to guarantee that if the corresponding input was in the 0..1 range, then the output is as well. In this case the user should be able to override the transformation being applied to the output.

The second approach is outlined first. It is based on the assumption that the worst case vector could occur. Consider an output vector **b** generated from the input vector **x** using $\mathbf{Ax}=\mathbf{b}$, with $0 \leq x_i \leq 1$.

Then $b_j = \sum_i a_{ji} x_i$.

The largest element of **b** can be no larger than

$$b_{\max} = \max_j \sum_{i, a_{ji}>0} a_{ji},$$

that is largest sum over a row, when only the positive elements are summed. The smallest element of **b** can be no smaller than

$$b_{\min} = \min_j \sum_{i, a_{ji}<0} a_{ji}.$$

After transforming, the maximum should map to 1, and the minimum to 0, so each element of **b** should be replaced with the corresponding element of $\mathbf{b}'$ where

$$b_i' = \frac{b_i - b_{\min}}{b_{\max} - b_{\min}} .$$

Note that the transformation is independent of **x** and **b**, as expected. This approach is device dependent, rather than image dependent. It is useful for presenting a series of images of varying contrast.

The other approach, that of transforming the input, is more involved. Here the maximum allowable input contrast must be found. Consider an output vector **b** generated from the input vector **x** using $\mathbf{Ax}=\mathbf{b}$. If **x** has no elements greater than $t_{\max}$ and no elements less than $t_{\min}$, then the greatest contrast that can appear in the input image is

$$\frac{t_{\min} - t_{\max}}{t_{\min} + t_{\max}} .$$

What values of $t_{\min}$ and $t_{\max}$ ensure that the vector **b** has no values outside the 0..1 range?

For row i of A there exists a vector $\mathbf{x}^{(i)}$, for which b_i is a maximum. It is defined as

$$x_j^{(i)} = \begin{cases} t_{\max}, & a_{ij} \geq 0 \\ t_{\min} & a_{ij} < 0 \end{cases} .$$

The vector $\mathbf{y}^{(i)}$ with roles of $t_{\max}$ and $t_{\min}$ reversed makes b_i as small as possible. That is,

$$y_j^{(i)} = \begin{cases} t_{\min}, & a_{ij} \geq 0 \\ t_{\max} & a_{ij} < 0 \end{cases} .$$

The smallest element of $\mathbf{b}$, over all allowable vectors $\mathbf{y}$, is $b_{\min} = \min_i \sum_j a_{ij} y_j^{(i)}$, while the largest is $b_{\max} = \max_i \sum_j a_{ij} x_j^{(i)}$. There are two important parameters of the output: the sum and difference of the max and min. The sum is

$$b_{\max} + b_{\min} = \max_i \sum_j a_{ij} x_j^{(i)} + \min_i \sum_j a_{ij} y_j^{(i)} .$$

The difference is

$$b_{\max} - b_{\min} = \max_i \sum_j a_{ij} x_j^{(i)} - \min_i \sum_j a_{ij} y_j^{(i)} .$$

When A is circulant, the sums do not vary from row to row.

$$\begin{aligned} b_{\max} + b_{\min} &= \sum_j a_{1j} y_j^{(1)} + \sum_j a_{1j} x_j^{(1)} \\ &= \sum_j a_{1j} (y_j^{(1)} + x_j^{(1)}) \\ &= \sum_j |a_{1j}| (t_{\max} - t_{\min}) \\ &= (t_{\max} - t_{\min}) \sum_j |a_{1j}| . \end{aligned}$$

$$\begin{aligned} b_{\max} - b_{\min} &= \sum_j a_{1j} x_j^{(1)} - \sum_j a_{1j} y_j^{(1)} \\ &= \sum_j a_{1j} (x_j^{(1)} - y_j^{(1)}) \\ &= \sum_j a_{1j} (t_{\max} + t_{\min}) \end{aligned}$$

$$= (t_{max} + t_{min}) \sum_j a_{1j} \ .$$

If $b_{min} = 0$, then $(t_{max} - t_{min}) \sum_j |a_{1j}| = (t_{max} + t_{min}) \sum_j a_{1j}$, so the maximum contrast is

$$C_{max} = \frac{t_{max} - t_{min}}{t_{max} + t_{min}}$$

$$= \frac{\sum_j a_{1j}}{\sum_j |a_{1j}|} \ .$$

If the device contrast is $C_d = \frac{b_{max} - b_{min}}{b_{max} + b_{min}}$, the maximum contrast in the image is

$$C_{max} = \frac{t_{max} - t_{min}}{t_{max} + t_{min}}$$

$$= \frac{b_{max} + b_{min}}{b_{max} - b_{min}} \frac{\sum_j a_{1j}}{\sum_j |a_{1j}|}$$

$$= C_d \frac{\sum_j a_{1j}}{\sum_j |a_{1j}|} \ .$$

This gives the maximum dynamic range under the worst possible conditions, assuming that the worst case vector for a given basis function might be specified as an input. Usually, more is known about the input so it is possible to do better. For example, the worst case vector may contain the max and min values in adjacent entries. If it is known that the image never changes that rapidly, the input may have greater dynamic range, without the output values leaving the allowable range.

The above analysis only gives a crude estimate of the dynamic range available. An eigenanalysis could also be used to find the worst case vector and the amount that the elements of that vector are scaled by multiplication by the matrix [Golub1983].

6.5. Constraints

Constrained minimization packages are readily available, for example in the NAG library [Numerical1988]. Details of the numerical methods are not relevant, but two features of them are worth noting. General (not hand-crafted) methods assume little or no structure exists within the matrix. Writing a hand-crafted method for a new kind of matrix structure, such as circulants dominated by the elements near the principal diagonal or near the off-diagonal corners, is a major undertaking. Second, adding constraints to a system is normally achieved by adding rows to the matrix and solving the new system. This eliminates the circulancy of the matrix, and makes it unlikely that a solution technique can take advantage of the original circulancy for improved efficiency.

Some methods require a starting guess for the solution, and here the circulancy may be used to compute a solution without constraints. If this solution does not violate any of the constraints, the problem is solved. Otherwise, the unconstrained solution provides a reasonable starting guess. One important reason for using this approach is that it can save a very costly calculation in the worst case. My experience, particularly with matrices used in frequency space minimization, has shown that if the starting guess derived in this way comes very close to violating the constraints without actually doing so, then the minimization routines, called with constraints, take days or weeks on a VAX 8600 to find a solution on moderate sized problems (16-20 samples). It is thought that this is as a result of all or nearly all of the constraints being active at each step of the solution process due to the oscillatory nature of the solution [Conn1989]. If the starting guess calculated using unconstrained techniques does not violate any constraints, then the calculation of the constrained solution is avoided entirely, since the unconstrained solution is satisfactory.

6.6. Fast Solution by Filtering

Each of the problems above has been cast into the form $\mathbf{Ax}=\mathbf{b}$ or $\mathbf{x}=\mathbf{A}^{-1}\mathbf{b}$. In each case, circulant techniques can be used to find $\mathbf{A}$ or $\mathbf{A}^{-1}$. It was already noted that multiplication by a circulant matrix is equivalent to numerical integration of a convolution integral, converging to the convolution as the size of the matrix gets large. In order to avoid side effects from assuming periodicity of the image, the matrix dimensions are increased until they are "large enough". Clearly no accuracy is lost in increasing the matrix dimensions further, producing a better approximation of the convolution integral. The only effect of this is to increase the computation time in performing the multiplication. Increasing the dimension to the next power of two allows the DFT used twice in the inversion of the circulant to be replaced by an FFT, with the result that the inversion takes time proportional to $n \log_2 n$, where n is the width of the matrix. Increasing the dimension indefinitely transforms the DFTs to

finite Fourier transforms, which are occasionally available in closed form. (Two inverse transforms are needed to solve a circulant system. It is quite possible that only one of them has a closed form solution.)

When the matrix is sufficiently large, most of its elements are effectively zero. Because the matrix is symmetric and circulant, the first off-diagonal is repeated in the top right and bottom left corners, which means that the matrix is not banded in the traditional sense. Nevertheless, the multiplication may be performed at a cost which does not escalate with the size of the matrix if we include only those elements which contribute non-negligible amounts to the sum.

If the problem is unconstrained, then it may be solved as follows: First find a large enough circulant or pseudo-circulant matrix to avoid the side effects of periodicity. This size depends only on the device and not on the image. Next invert the matrix, or find a pseudo-inverse in the case of a pseudo-circulant matrix, to produce a new matrix, either circulant or the transpose of a pseudo-circulant matrix. This inverse is the filter, sampled at all the points at which it will be needed for the convolution. Lastly, convolve the input with the filter. This involves, for each output pixel, as many multiplications as there are non-negligible entries in a row of the matrix, and one fewer addition. The total cost is proportional to the size of the output, unless it is so small that the cost is dominated by the inversion of the matrix. Because the filter is based on unconstrained mimimization, values outside the device range may be generated. Two solutions are suggested in Section 6.4. An *ad hoc* solution is also possible, clipping the output to the device range. If the solution is very close to fitting within the device range, then clipping can be justified on the basis that the changes which result are no larger than the round-off error in the solution. Alternatively, constrained minimization, outside the filtering paradigm, may be performed.

Simulation Techniques

Given a device with known characteristics (pixel shape and phosphor decay time), it is easy to display the results of a minimization on that display, as long as the device characteristics assumed in the minimization are the characteristics of the device. This is often too limiting. It is much more interesting to be able to compare different minimizations on a single device, and the effects of different devices. Without acquiring devices with different characteristics, it is possible to simulate, in a limited way, the characteristics of many devices using a high quality CRT. This has the advantage that it is possible to simulate devices which do not even exist. It also keeps all device characteristics other than pixel shape (such as dynamic range, colour balance, etc.) constant. This chapter describes simulations that have been done using an Adage/Ikonas frame buffer and various multisync monitors. These simulations allowed experimental validation of the theory.

The simulation can be carried out in either the spatial or the temporal domain. The calculations are identical, but the details of display and viewing differ. Calculation methods are presented first, and then details of display are given. When image pixels are simulated using multiple device pixels, in principle any image pixel shape can be used. Some shapes are more useful than others. A criterion for finding the best pixel spread for any one-parameter family of pixel shapes is given in the last section of this chapter. For Gaussian pixels this leads to a particularly simple relationship between the pixel half-width and the inter-pixel separation.

7.1. Simulation in general

To see the effect of a different pixel basis function, compute the image for a low resolution device, then at as many sample points as there are pixels on the actual device, use the methods of Chapter 6 to compute the intensity which the simulated device would produce. This gives a set of samples of an image to display on the actual device. Since the samples are at the same frequency as pixels in the device, the matrix involved in the methods of Chapter 6 is symmetric and circulant, which means that the solution may be obtained quickly.

It is often the case that the actual device pixel basis can be approximated in simulations as a δ function, with the simulated basis having a spread over several device pixels. In temporal simulations this is the case whenever the decay of the phosphor is fast enough that a negligible amount of light remains from the previous actual frame when the next frame occurs. This criterion depends on the actual frame rate. In spatial simulations it is safe to treat device pixels as δ functions if they are viewed from sufficiently far away. If the device pixels subtend a visual angle at or below the limit of visual acuity, then their intensity profile has no effect on their appearance. (This corresponds to roughly one minute of arc).

In Chapter 6, several methods of finding the best pixel values for a given device are discussed. Of these, the direct method uses a matrix **A** which, when multiplied by a vector of pixel values **x**, gives a vector of intensities **b**. To find the best pixel values, $\mathbf{Ax}=\mathbf{b}$ is solved for **x**. Here the reverse computation is performed: **A** is multiplied by **x** to give **b**. In this chapter it is assumed that a vector of pixel intensities for the simulated device is already known, and the problem is to display it.

Computationally, it doesn't matter whether spatial or temporal profiles are simulated. Let the intensities of the image pixels be specified as x_j, and the intensities of the device pixels be b_i. Assume that the values of x_j have been specified for enough points beyond the image that the introduction of periodicity does not produce any errors, and that the period is then fixed at $T = M\tau = N\tau'$, where τ and τ' are the inter-sample and interpixel separations, respectively. The intensity at $t = i\tau$ is given by

$$I(i\tau) \equiv b_i = \sum_{p=-\infty}^{\infty} \sum_{j=1}^{M} x_j B(j\tau' - (pN + i)\tau) \quad .$$

Until $B(t)$ is specified, this is the general transformation required for simulating an

arbitrary pixel basis with δ functions. It may be re-cast as $\mathbf{b}=\mathbf{Ax}$, with $\mathbf{A}$ circulant if $N=M$, and pseudo-circulant if $M<N$. Normally the number of pixels is substantially greater than the number of samples, so that $\mathbf{A}$ is only pseudo-circulant.

$$A_{ij}=\sum_{k=-\infty}^{\infty} B(j\tau'-(Nk+i)\tau),\ 1\leq i\leq M,\ 1\leq j\leq N\ .$$

To compute a scanline spatially or a series of values at one position for a sequence of frames, find the matrix $\mathbf{A}$ and multiply the vector of device intensities by it. The matrix only needs to be calculated once for the device/resolution combination, and not for each new input image. If the matrix is treated as a filter (as in Section 6.6), some multiplications may be avoided.

Simulation in one dimension may be extended to two dimensions straightforwardly for any separable pixel shape, of which the Gaussian of a CRT is a good example. Begin with a frame buffer of values to be displayed on the simulated CRT. Each horizontal scanline of the actual frame buffer is computed independently, using several scanlines of the frame buffer for the simulated device. This gives an intermediate image which has a vertical resolution corresponding to the simulated device and horizontal resolution corresponding to the actual device. Then each vertical scanline of the final frame buffer is calculated in the same fashion, yielding the final set of pixel values for the actual CRT. The two passes may be combined into one by keeping a small number of intermediate scanlines in memory and computing output scanlines as soon as enough information is available. If the simulated pixel shape is not separable, the two dimensions must be treated in one pass, but the problem is conceptually the same.

7.2. Spatial Simulation

Spatial simulation is a relatively easy task, requiring no special features of either the monitor or the frame buffer. To give control over the shape of the pixels, each pixel is built up from many actual device pixels. The more device pixels used, the better the control, but the lower the resolution of the image.

If enough device pixels are used per image pixel, then it is safe to approximate a device pixel as a δ function. If relatively few device pixels are used per image pixel, then the shape of the device pixels must be taken into account. Figure 10 shows a single Gaussian image pixel simulated using multiple Gaussian device pixels. Here seventeen device pixels make up a single image pixel, with (overlapping) image pixels spaced 5 device pixels apart. The device pixels are approximated as δ functions for the calculation of their amplitudes.

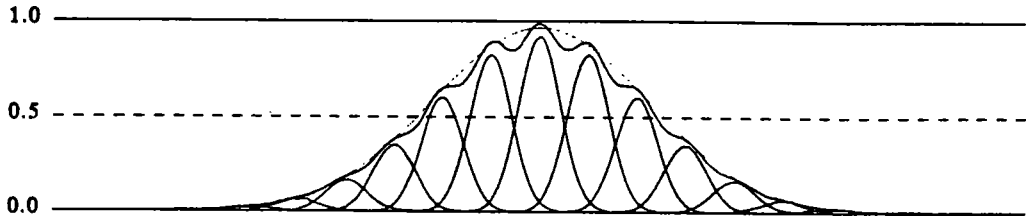

Figure 10: One Gaussian simulated using multiple Gaussian device pixels. The pixel spread is appropriate for an inter-pixel separation of five device pixels. The smooth curve at the top is the ideal Gaussian, the wavy one below it is the actual simulated pixel. The waviness of the simulated pixel is small enough to be invisible from the proper viewing distance.

7.3. Temporal Simulation

Temporal simulation is the same as spatial simulation except that Gaussian actual pixels are replaced by exponential ones. Fortunately monitors with short persistence phosphors are available, and their persistence is sufficiently short that it is a good approximation to treat the temporal pixel shape as a δ function. It is necessary to have a high frame rate device to simulate normal frame rates so that multiple actual frames combine to produce one simulated frame.

By running the monitor at a high enough frame rate, a reasonable approximation of an image pixel shape may be simulated, where the image pixel is replicated at a normal frame rate. Since the visual system is practically blind to temporal frequencies higher than 60 Hz, one choice of image pixel is visually identical to all other possible image pixels which differ only in their high frequency (>60 Hz) content. To avoid aliasing, the monitor is driven well above 120 Hz (non-interlaced), so that a sampling of an arbitrary shaped 30 Hz pixel at the frame rate of the monitor is indistinguishable from a continuous representation. The value of 30 Hz is chosen for two reasons: first, it is a commonly used refresh rate in many display systems; second, it is sufficiently slow that the effects of changing pixel profile are easily visible.

7.4. Display

To simulate arbitrary temporal pixel shapes a monitor capable of high frame rates, and a frame buffer capable of driving it are needed. The Adage/Ikonas 3000 is one of few frame buffers capable of such line rates. Even on the Ikonas, these line rates are only possible as long as the number of pixels read from memory in a frame time does not exceed the memory bandwidth. Fortunately, the simulation is equally informative with only a few (100 or fewer) lines displayed on the monitor. The image is computed as $I(x,t)$. with all of the lines the same. Multisync monitors capable of more than 150 fields per second are widely available. These simulations used a Sony

CPD 1302 monitor and a Gigatek CCD1331ST/AC/HG monitor (no longer in production).

The Ikonas video chain consists of frame buffer memory, a parameterized video controller with 8 control registers, a crossbar switch, colour maps, and digital to analogue converters. Pixels are read from the frame buffer in scanline order and their values are passed through the crossbar switch to the colour maps. The crossbar switch allows arbitrary connections of the 32 bits of a pixel from the frame buffer to the 32 address bits used in the colour map. Any of the crossbar switch bits may be disconnected under program control. The most common setting is straight through: every input bit maps to the corresponding output bit. The colour map provides a mapping from pixel values received from the crossbar switch to voltages at the digital to analogue converters.

Among the frame buffer controller registers are the window location (pan/scroll), zoom, display rate control and mode registers. The mode register consists primarily of bit fields controlling various modes of the Ikonas operation. The window location controls the location in memory of the first pixel to be read each frame. If the zoom register is non-zero, then pixels are replicated in x and y the indicated number of times. The display rate control register may be adjusted to provide non-standard frame rates. The x field determines the number of 209.5ns clock cycles in each scanline, and the y field indicates how many scanlines to display before vertical retrace. The mode register contains a number of fields, among them a bit which selects internal or external sync (internal is required here), and one which selects between repeat field and interlaced mode. Repeat field causes the same image to be displayed twice, in the same place, instead of two images displayed alternately using interleaved scanlines. Unless compensation is made for the effects of interlace, the second image appears to be lagging behind the first, resulting in a ragged appearance of vertical edges moving rapidly in the horizontal direction. Repeat field mode provides the compensation. Further details are available in the Ikonas programmers manual [Adage1982].

Values of (303,187) (decimal) in the display rate control register give a frame rate of approximately 168 Hz, without loss of sync on the Gigatek monitor. This is well above the visual cutoff frequency, so that frames constructed from 5½ actual frames appear to be frames at about 30 Hz with the basis function used for the construction.

The object displayed in animated sequences is a single square or vertical bar moving across the screen. This is representative of a wide class of objects encountered in computer graphics since it contains two moving edges, one "rising" and one "falling". It is not actually free of high frequency information, but a sampling of it is

also a sampling of a different object which is free of high frequency information. An ideal square wave has significant power at high spatial frequencies; the high frequency power is necessary to produce square corners. However, samples from a square wave are close to samples from a similar ideal image without high frequency information, if the square wave has a wavelength which corresponds to an integral number of pixels, and its rising edge is pixel-aligned.

It can be modelled either as a two dimensional function $I(x,t)$ or a series of one dimensional functions $I_x(t)$ with $I_{x+1}(t)=I_x(t+v)$. In the latter case, the frame buffer is loaded with a linear ramp of values, so that the contents of the jth column is j, and then the window x register is incremented by one each frame time so that the contents of colour map register $i+j$ are displayed in column j at frame i. (This gives a maximum of 256 frames per cycle). Colour map register i is loaded with the (gamma corrected) intensity value for frame i. This display method has two advantages: $I(t)$ may be changed rapidly by re-writing the colour map, and colour map entries may be calculated to a full 10 bits, customized for $I(t)$.

Since every scanline is the same, the zoom register is used to replicate scanlines, up to a replication count of 16 (256 on older models of the Ikonas). If only 187 scanlines are displayed before vertical retrace, only 12 unique lines need to be supplied per frame. They can be arranged in the frame buffer in successive positions in memory, 3 across, and 85 down, for a total of 255 frames in a 1024×1024 frame buffer. Then the window register x and y fields are changed each vertical retrace time to display successive frames. Greater control over $I(x,t)$ is achieved at the cost of having a fixed colour map, and slower update rates when the parameters of the simulation are changed.

In visual comparisons between the results of different techniques, the screen is blank while a new $I(x,t)$ is being computed. It is easier to see the transition between two (spatiotemporal) images than it is to see the difference between them. If they are indistinguishable, then the techniques should be considered equivalent. It may be that a transition is visible. The analogous situation is computing two (static) periodic images and then asking an observer whether they are the same. If the observer is permitted to place them right against each other, lined up as to phase, then he is able to detect much smaller differences by looking for an edge between them than if they are to be viewed in different positions with a slight gap between. For temporal images, the equivalent of the gap is a blank screen. The fastest way to blank the screen is to set the crossbar registers to unconnected. This has the advantage that it affects the contents of neither the frame buffer nor the colour map.

If the intensities calculated are the not same as the intensities produced by the monitor, the entire exercise is meaningless. Gamma correction is needed to compensate for the nonlinearity of the function relating gun voltage (which is linear in colour map value) to intensity. A good first approximation uses $I \propto V^{\gamma}$, hence the name gamma correction [Foley1983]. Catmull discusses more accurate device calibration [Catmull1979]. One way or another, the correspondence between values stored in the colour map and values of intensity perceived or measured is found and tabulated, and then appropriate values are stored in the colour map to compensate.

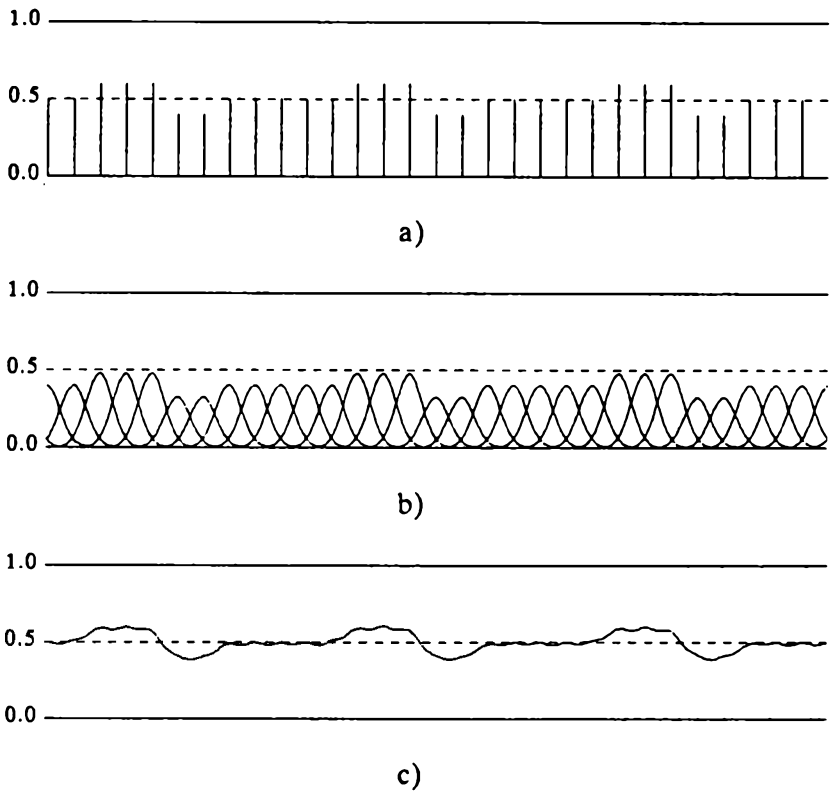

Figure 11: An image drawn with Gaussian pixels: a) frame buffer values, b) individual pixels, c) resulting intensity profile. The low amplitude ripple is not visible at normal viewing distances.

7.5. Pixel Shape

Since any pixel shape can be simulated, there is the question of which shape is better than another. In this section, the decay parameters of the two parts of the CRT pixel basis are explored.

In the temporal domain, the choice of decay parameter is a compromise between flicker and loss of contrast. In the spatial domain, the compromise is between a loss of flat field and a loss of definition. A sum of very broad Gaussians is effectively flat, while a sum of narrow Gaussians is like a set of translated δ functions. If the decay parameter is too large, then edges are excessively blurred; if it is too small, scanlines are visible. One other criterion is important: there should be a minimum of information above the fundamental frequency in an image of alternating scanlines. Since any antialiased image contains no information above the fundamental frequency in an image of alternating scanlines, any information in such an image is introduced by the pixel shape and results in artifacts in all images produced on the device.

For Gaussian pixels, the basis function is $B(t) = \frac{\delta}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2}$. This function is normalized so that an infinite series of such pixels, all with unit values and spaced δ apart, have unit integrated intensity in a unit interval. Figure 11 shows an image displayed with the best value of σ. A slight ripple appears in the diagram in the parts of the image corresponding to constant values in the frame buffer. This ripple is small enough that it is not visible in the image. Figures 12 and 13 show the effect of changing σ. For larger values of σ the ripple disappears, but edges are blurred. For smaller values of σ the ripple is visible. The choice of σ cannot be made without knowledge of the human visual system.

Figures 14 and 15 show the pixels and the resulting intensities respectively for an exponential decay. Where the decay rate is fast, flicker is a problem, where it is slow, flicker is not a problem, but the intensity cannot fall as quickly from one frame to the next. An exponential decay is considered because it models the temporal behaviour of CRTs. But if the device being used has high enough resolution to permit arbitrary pixel basis functions to be simulated, the exponential decay is not the basis of choice. A Gaussian pixel basis is more useful than the exponential for several reasons. First, it has much faster asymptotic decay. Secondly, it has no discontinuities, which means that less high frequency information is introduced by the pixel profile. The symmetry of the Gaussian basis is also useful. For these reasons a Gaussian pixel was simulated both temporally and spatially.

As already mentioned, the ideal pixel basis is a compromise. The best values of the parameter are those for which the ripple between scanlines or the flicker between frames is just barely invisible. Both intense flashes and a flicker-free static image are

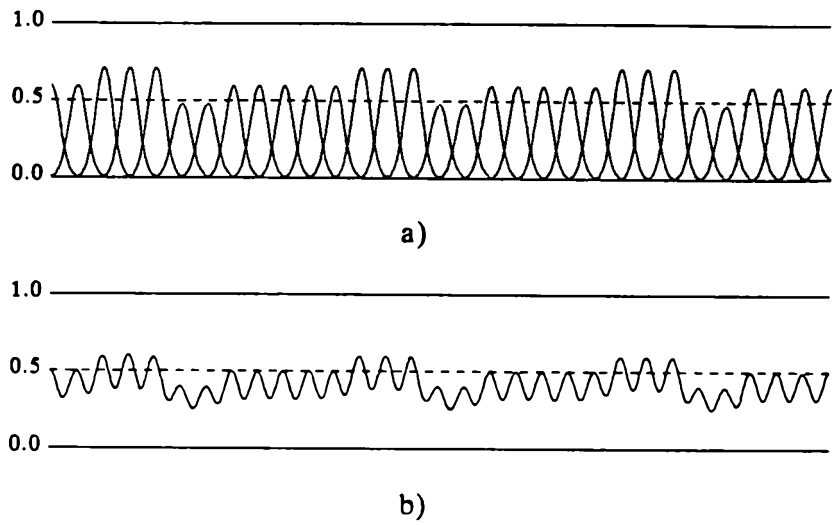

Figure 12: The effect of reducing the Gaussian parameter to 2/3 of its value in Figure 11. The framebuffer values are unchanged.

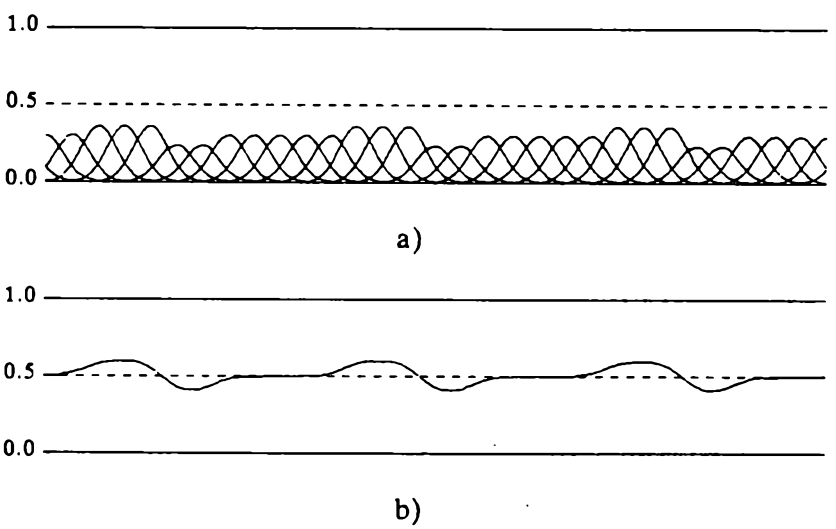

Figure 13: With the Gaussian parameter at 4/3 of its value in Figure 11, the ripple is gone, but edge sharpness is lost as well.

possible when the temporal simulation is done correctly. The choice of parameter depends on the spatial and temporal frequencies involved, just as the threshold of visibility depends on the frequency. Under "normal" room lighting, the appropriate value of the Gaussian parameter was nearly linear in the separation of the peaks, when measured by two observers in both a grating visibility experiment and a flicker experiment, as shown in Figures 16 and 17. From these graphs, a good approximation, for most frequencies of interest, is $\sigma/\tau = .41 \pm .01$ with τ the inter-

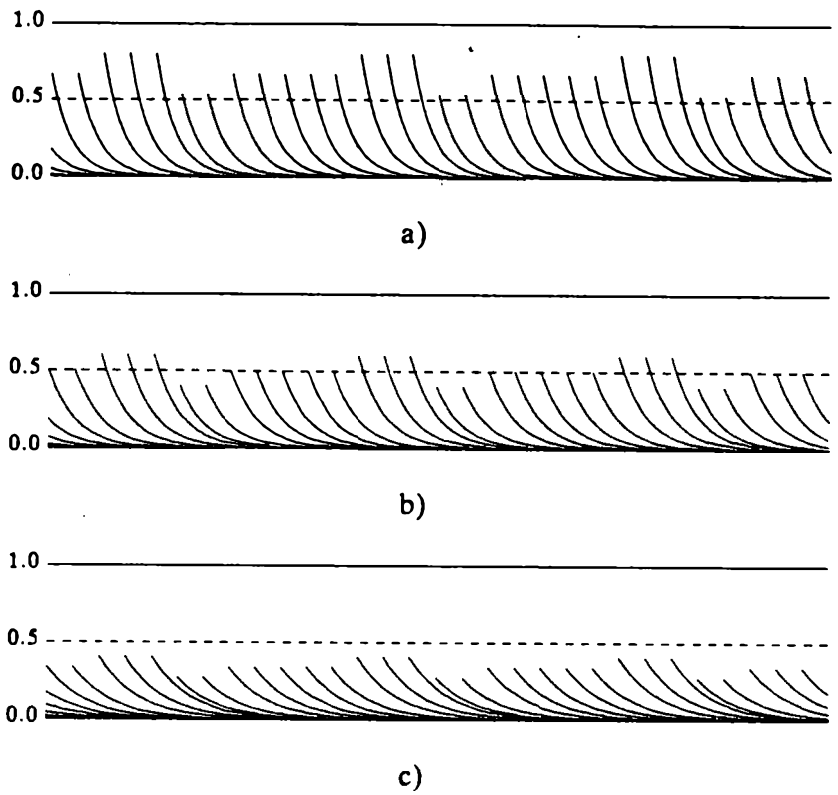

Figure 14: The effect of changing the value of the exponential parameter on image construction. Here individual frames are shown with their entire decay curve. This corresponds to part a) of Figures 11-13. The three sequences appear equally bright; a) is more likely to show flicker, c) is likely to have after-images. The intensities resulting from summing the intensities of all frames, corresponding to part b) of Figures 11-13, are shown in Figure 15.

frame separation in the temporal case, and $\sigma/\delta = .51 \pm .01$, with δ the inter-pixel separation. Kajiya and Ullner state that the appropriate spatial Gaussian parameter is given by $\sigma/\delta = .43$, but give no supporting evidence [Kajiya1981]. The linearity is remarkable. The contrast of the simulated flat field is a non-linear function of σ; the minimum visible contrast is a non-linear function of spatial frequency, which is the inverse of inter-pixel separation; yet for a wide range of spatial frequencies the value of σ corresponding to the minimum visible contrast is linearly related to the inter-pixel separation.

Another way of finding the value is to note that each of the images produced as a sum of Gaussians is well approximated by a sinusoidal ripple added to a large constant brightness. (The approximation fails when σ is too small, but then non-uniformity or flicker is clearly visible.) An harmonic analysis reveals that the energy at any frequency other than the fundamental is less than 10^{-4} as much as that at the

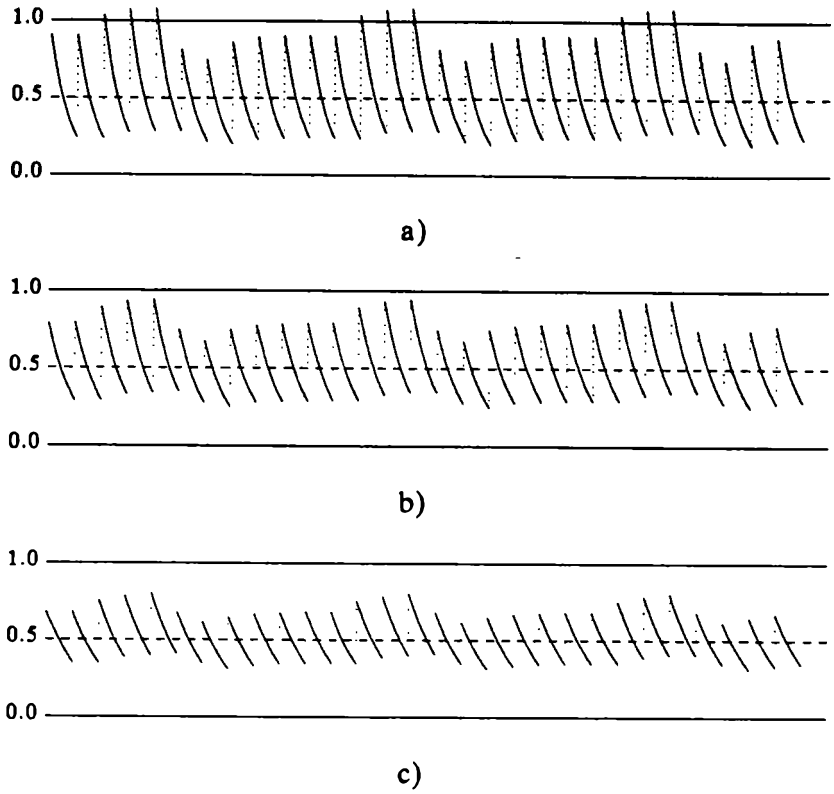

Figure 15: The actual intensities corresponding to the frames in Figure 14. Because frames fade more rapidly in a), the intensity may be dropped more rapidly without an after-image appearing. At the same time, flicker of a static image is most likely to be a problem in a). The lag between a transition in the frame buffer value and a stable intensity is indicative of the need to compensate for the device.

fundamental. If the contrast of the fundamental is small enough to be an invisible deviation from a flat field, then the image appears flat, since the other terms are not visible. The best pixel shape has the largest possible contrast in the fundamental frequency without losing the effect of a flat field.

The data in Figures 16 and 17 are the raw data from experiments done as part of this research which measured the minimum value of σ for flat and flicker-free fields, respectively. The contrast may be computed from the sum of a small number of terms in the series expansion for the intensity. The brightest value occurs at the centre of a pixel or start of a frame, the dimmest midway between. The difference between the dimmest and the brightest divided by their sum gives the contrast. Graphs of derived contrast vs. spatial and temporal frequency appear in Figures 18 and 19. They show a marked similarity to Robson's data (Figures 5 and 6 in Chapter 3). Since the experimental conditions are less controlled, the errors are larger than those in Robson's data, but the trend is the same. Thus Robson's data may be used

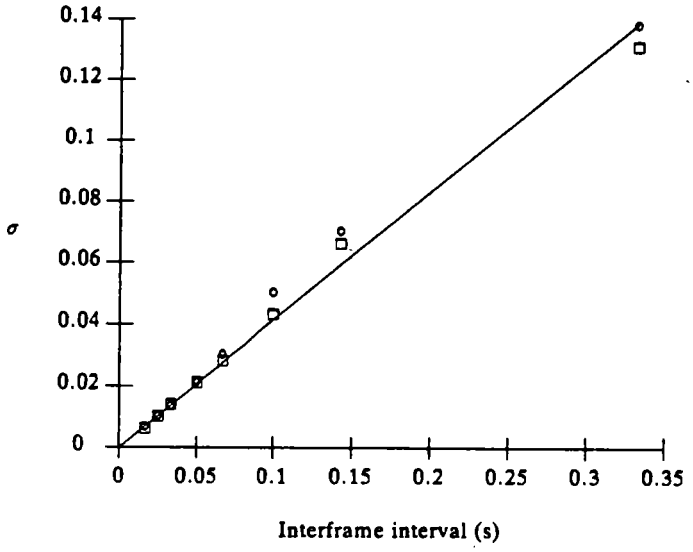

Figure 16: Minimum flicker free Gaussian parameter, as a function of inter-frame separation.

directly in finding the appropriate value of the Gaussian parameter. The method is applicable to other one-parameter basis functions as well: find the mapping from the parameter to contrast in a simulated flat field, and select the parameter value which puts the contrast at the threshold of the human visual system for the temporal and/or spatial frequency of the device.

Computational note:

The sum of basis functions needed to compute the image at the pixel centre and midway between may be rewritten as $\sum_{k=-\infty}^{\infty} B((j\tau - i\tau') - Nk\tau')$ to emphasize the part which is held constant. It may be calculated efficiently by finding the range of k for which terms are sufficiently large to have a significant effect on the image, and then omitting terms outside of this range.

For a Gaussian pixel, let ϵ be considered negligible as a term in the series. Then if $x = j\tau - i\tau'$, a term is negligible when $e^{(x-Nk\tau')^2/2\sigma^2} \leq \epsilon$ or $(x-Nk\tau')^2 \leq 2\sigma^2 \ln\epsilon$. Solving for k gives the values of k beyond which terms need not be computed.

$$(Nk\tau')^2 - 2xNk\tau' + x^2 - 2\sigma^2\ln\epsilon = 0 \Rightarrow k_0 = \frac{x \pm \sigma\sqrt{-2\ln\epsilon}}{N\tau'}$$

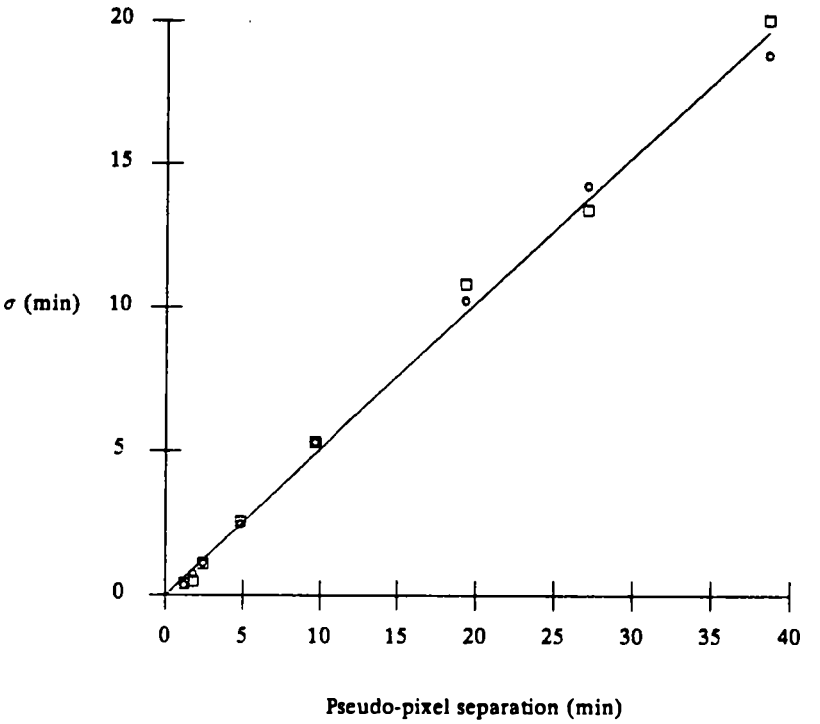

Figure 17: Minimum Gaussian parameter retaining a flat field, as a function of pixel separation.

$$\frac{x-\sigma\sqrt{-2\ln\epsilon}}{N\tau'} \leq k \leq \frac{x+\sigma\sqrt{-2\ln\epsilon}}{N\tau'} .$$

Note that $x/N\tau'$ is always between -1 and 1, and $\epsilon \ll 1$.

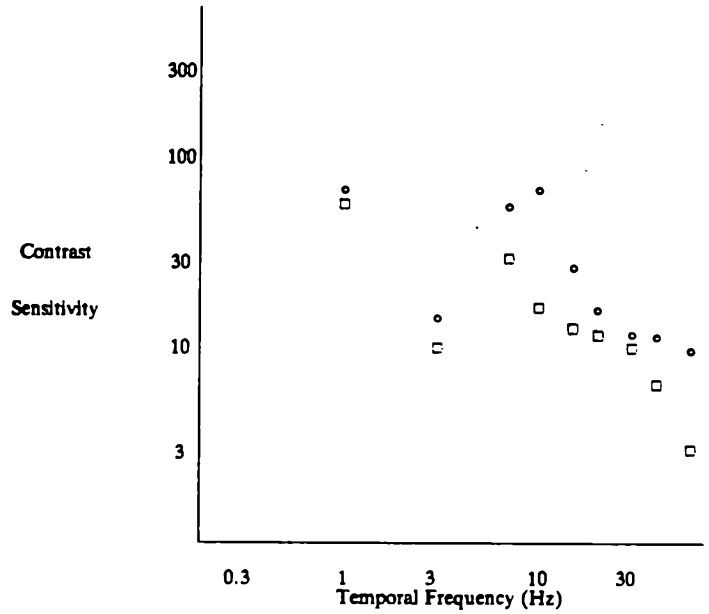

Figure 18: Contrast threshold derived from the data of Figure 16, plotted on the same scale as Robson's temporal contrast threshold data (Figure 5).

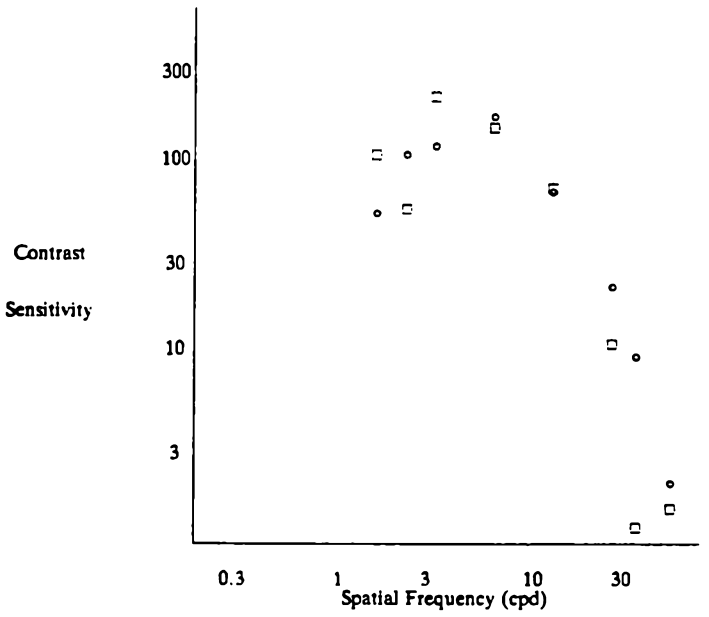

Figure 19: Contrast threshold derived from the data of Figure 18, plotted on the same scale as Robson's spatial contrast threshold data (Figure 6).

One Dimensional Image Construction

In this chapter two examples of finding the best sequence of pixel values for a (one dimensional) temporal or spatial image are presented. The computational techniques have already been discussed; here they are applied to two actual pixel shapes, for a single image. The two pixel shapes are the exponential decay and the Gaussian. The exponential basis is important because it describes the temporal profile of pixels on a CRT, the most common graphical display device. Some LCDs also have an exponential decay. The Gaussian is a good approximation to the vertical spatial profile of a pixel on a CRT. An idealized CRT is one where the duration of the beam intensification is vanishingly small. On such a device, the Gaussian is a good approximation to the horizontal spatial profile as well. The image is a square wave, bright one half of the period and dim the other.

For each of the two basis functions, the matrix used in direct minimization is derived. In addition the frequency space solution is described for the exponential basis. This solution is band-limited, since the Fourier series representation of the square wave is used for the ideal image. The consequences of using various choices of norm are also discussed.

Currently no device has a Gaussian temporal basis function, but it is conceivable that future devices may simulate arbitrary pixel bases using specialized hardware. This applies both in the temporal and spatial domains. Among possible customized pixels, symmetric ones are likely to be desirable, just because equal numbers of images are likely to be well suited to pixels skewed in either direction. Of symmetric pixels, the Gaussian is an example which is non-negative, while introducing little spurious high frequency information.

The solutions to the minimization at the edges of a square wave indicate how the edges of polygons should be treated in a static image. The temporal intensity profile of a single (spatial) point in an image containing a moving object may be better described by a pulse with sloped sides, but the methods used to compensate for the device are no different. A square wave does not meet the criterion for lack of coherent high frequencies; a sampling of a square wave at the pixel rate does. (It isn't

exactly a band-limited square wave, but there is a band-limited function that interpolates the sampling, by Whittaker's theorem [Whittaker1915] - see also the Appendix.) A band limited square wave is also constructed using the Fourier series representation. When sampling above the pixel rate, aliasing is thereby eliminated.

8.1. Exponential decay

The exponential function has a property which makes the direct solution particularly simple. The sum of two exponentials of different starting values but the same decay rate is also an exponential of the same decay rate. Thus, the total intensity in the interval between two pixels is an exponential of the same characteristic decay rate. As a consequence, only the total intensity remaining from the previous pixel and the intensity required at the current pixel need be considered in computing the value for a given pixel.

Of direct methods, the solutions obtained by matching the integrated intensity value over each pixel are in general better than those obtained by matching samples. As shown in Section 6.2, matching integrated intensity values is equivalent to matching the intensity values at each pixel, using a different basis function. Because the integral of an exponential is a constant times an exponential of the same decay rate, the basis used for matching samples is the same as that used for matching integrals, except for a normalization constant. Matching the values at each pixel is simpler, and in this case is nearly the same. A closed form solution exists for unconstrained minimization using exponential basis functions. This solution requires limiting the contrast in the input (ideal) image if negative pixel values are to be avoided. When the number of sample image values specified is greater than the number of pixels required, the problem is complicated by the choice of norm. In frequency space, there are also several norms available, although the l_2 norm solution may be found much more quickly.

8.1.1. Pointwise Matching

Assume that the desired image is specified at regularly spaced sample points with the same period as the pixel spacing, τ. Without loss of generality, assume that the image repeats every N pixels. If the image is not inherently periodic, blank pixels can be appended to remove any spillover of intensity from one repetition to the next. The input sample values also repeat every N samples. They are given by the elements a_j of a vector $\mathbf{a}$ of length N.

In this thesis pixels are defined in space and time. The exponential decay is common in temporal profiles of pixels, but not in spatial profiles. It is not even desirable in spatial profiles, due in part to the inherent assymetry. In the remainder

of this section, pixels are synonymous with frames, and pixel profiles are temporal. To make it easier to visualize the effects described, "time" is explicitly used, rather than "space or time".

Time $(j-1)\tau$ is the beginning of interval j and time $j\tau$ the end. The normalized exponential basis is $\lambda\tau e^{-\lambda t}$ so the contribution of pixel j at the start of the jth pixel is

$$a_j \lambda \tau (1 + e^{-\lambda N \tau} + e^{-2\lambda N \tau} + \cdots)$$

(1 for the pixel that just started, $e^{-\lambda N \tau}$ for the pixel a period previous, and so on).

$$= a_j \lambda \tau \sum_{k=0}^{\infty} e^{-k \lambda N \tau}$$

$$= \frac{a_j \lambda \tau}{1 - e^{-\lambda N \tau}} .$$

Usually, $e^{-\lambda N \tau}$ is negligible. More generally, to find the contribution of a_i to the intensity at pixel j, there are two cases:

1) For $0 \leq i \leq j$, the contribution from pixel i to the intensity at the start of pixel j, is its contribution to pixel i reduced by the decay between pixels i and j.

$$\frac{a_i \lambda \tau}{1 - e^{-\lambda N \tau}} e^{-\lambda (j-i) \tau} .$$

2) For $j < i < N$, the contribution from pixel i is the contribution to pixel $i - N$ reduced by the decay between pixels $i - N$ and j.

$$\frac{a_j \lambda \tau}{1 - e^{-\lambda N \tau}} e^{-\lambda (j-i+N) \tau} .$$

Thus the total intensity at the start of pixel j is given by $I(j\tau) = \sum_{i=0}^{N-1} a_j b_{ij}$, with

$$b_{ij} = \begin{cases} \dfrac{\lambda \tau}{1 - e^{-\lambda N \tau}} e^{-\lambda (j-i) \tau} , & 0 \leq i \leq j \\ \dfrac{\lambda \tau}{1 - e^{-\lambda N \tau}} e^{-\lambda (j-i+N) \tau} , & j < i < N \end{cases} .$$

The matrix **B** is circulant, from Theorem 5. Thus rapid solution techniques, requiring only $O(n \log n)$ time, can be used to solve $\mathbf{Ba} = \mathbf{I}$ for $\mathbf{a}$. Figure 20 shows the resulting intensities when the right-hand-side vector is specified as eight pixels of intensity 0.7 followed by eight pixels of intensity 0.3.

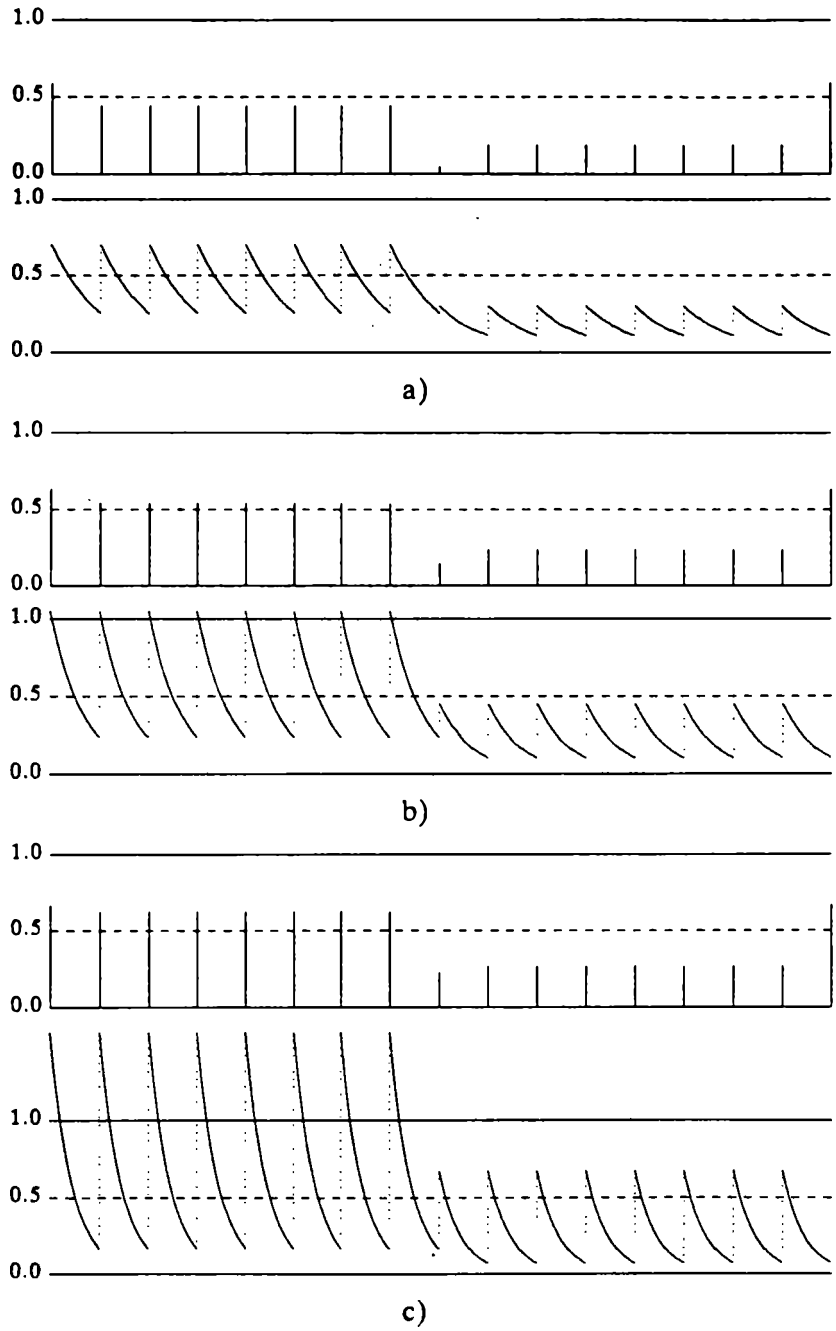

Figure 20: A square wave of amplitude 0.2 corrected for an exponentially decaying pixel. $\tau = 1/16$, $\lambda =$ a) 16, b) 24, c) 36. In each pair the bottom picture shows the intensities generated by the values in the top picture.

In the figures the integrated intensity in each pixel is matched to the integrated intensity of the square wave over the same interval. The top picture in each pair shows the frame buffer values; the bottom one shows the intensities that result. Note from the frame buffer values that exactly one pixel of overshoot is required at each transition. The intensity values are stable immediately after the transition.

If the sampling rate and pixel rate are different, **B** is pseudo-circulant. If the pixels are spaced τ' apart, the two cases (on page 80) become

1) $j\tau' \geq i\tau$:

the contribution is $\dfrac{a_i \lambda \tau}{1-e^{-\lambda N \tau}} e^{-\lambda(j\tau' - i\tau)}$

and $b_{ij} = \dfrac{\lambda \tau}{1-e^{-\lambda N \tau}} e^{-\lambda(j\tau' - i\tau)}$.

2) $j\tau' < i\tau$:

the contribution is $\dfrac{a_i \lambda \tau}{1-e^{-\lambda N \tau}} e^{-\lambda(j\tau' - i\tau + N)}$

and $b_{ij} = \dfrac{\lambda \tau}{1-e^{-\lambda N \tau}} e^{-\lambda(j\tau' - i\tau + N)}$.

With more samples than pixels the problem is overdetermined. In such a case it is important that the image be band-limited. The need for band-limiting in the overdetermined case is discussed further in Section 8.2.

Solving the circulant equations is appropriate in general cases. The exponential pixel has special features that make it easy to work out a solution in closed form. The figures show values which are constant after one frame after a transition either from high to low or the reverse. This is no accident. As the following derivation shows, only the first pixel after a transition needs to be adjusted to compensate for the transition.

Consider the trailing edge of a square wave, positioned at $t = 0$. The exponential pixel produces light intensity that decays as $Ae^{-\lambda t}$. The first pixel after the square wave occurs at $t = \tau$. If this pixel is given a value of $-Ae^{-\lambda\tau}$ then it contributes $-Ae^{-\lambda\tau}(e^{-\lambda(t-\tau)}) = -Ae^{-\lambda t}$ to the signal, exactly cancelling the decaying intensity from the square wave for all $t > \tau$. Negative intensities are not possible, but if the transition is to a non-zero value $C \geq Ae^{-\lambda\tau}$ the pixel may be given the value $C - Ae^{-\lambda\tau}$ with all following values at C. The subtracted quantity $-Ae^{-\lambda\tau}$ exactly cancels the surplus for all time.

This procedure is satisfactory provided the contrast at the transition is low enough. In the next section, the limits to the contrast are derived. The derivation uses a square wave of period four pixels. This is enough, because of the four pixels, two are at transitions, and two are at the steady state level. In the remainder of this

section, the interrelationships between the four pixels are derived.

Let J_i be the intensity setting for pixel i, $i=0,1,2,3$ and I_i be the intensity value resulting from this setting and any remaining intensity from all previous pixels. Then to obtain the square wave, set $J_0=1, J_2=0$, and $I_0=I_1$, $I_2=I_3$. The solution is the square wave of maximum amplitude for a given pixel basis. The decay of the exponential dictates that $I_i=I_{i-1}w+J_i$, with $w=e^{-\lambda\tau}$. There are 8 equations in 8 variables, where I_0, I_2, J_1 and J_3 are interesting. The solution of the system yields

$$I_0=\frac{1}{1-w^2} \quad I_1=\frac{w}{1-w^2}$$
$$J_1=\frac{1}{w+1} \quad J_3=\frac{w}{w+1}. \tag{14}$$

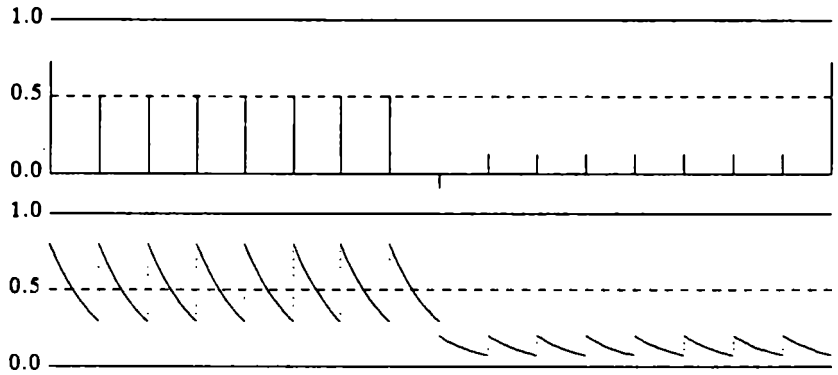

Figure 21: With $\lambda\tau=1$, a square wave of mean 0.5 and amplitude of 0.3 requires negative intensity values.

Note that in Figure 20 as the persistence decreases (λ increases), the amount of correction required decreases as well. If the amplitude is increased to 0.3, with the mean at 0.5, the solution for $\lambda\tau=1$ requires negative values. Figure 21 shows such a case. (If the pixels are simulated from higher frequency pixels with a very short decay constant, negative values can be simulated, since the total intensity is never negative). The amplitude at which negative values first appear specifies the available contrast.

8.1.2. Available Contrast

From Figures 20 and 21, it is clear that when the contrast is too great (as in Figure 21), negative intensities are required. For shorter persistence phosphors (Figure 20 b and c), the contrast may be greater than for longer persistence phosphors. Given the intensities obtainable for a square wave (14), it is easy to calculate the available contrast:

$$contrast = \frac{I_0 - I_2}{I_0 + I_2} = \frac{(1-w)/(1-w^2)}{(1+w)/(1-w^2)} = \frac{1-w}{1+w}$$

$$\text{As} \quad \lambda\tau \rightarrow 0, \quad \text{then} \quad w \rightarrow 1 \quad \text{and} \quad contrast \rightarrow 0$$

$$\text{As} \quad \lambda\tau \rightarrow \infty, \quad \text{then} \quad w \rightarrow 0 \quad \text{and} \quad contrast \rightarrow 1$$

Manufacturers have built two kinds of monitors: one type has long persistence phosphors and the other has short persistence phosphors. Long persistence monitors are designed for display of static images, particularly computer generated static images. These monitors have no perceivable flicker. Flicker is a particular problem for images containing high contrast one pixel high horizontal lines. When these are displayed at 30 Hz interlaced, the image is blank for alternate fields and the display flickers at 30 Hz. With long enough persistence phosphors the flicker is not visible. Short persistence monitors are designed for display of moving images, such as television pictures. Images broadcast for television viewing generally have little high contrast detail so that flicker is less of a problem.

Long decay times correspond to low values of λ. Low values of λ reduce the available contrast. The other way to reduce flicker is to drive the monitor at a high enough frame rate that the flicker is invisible. (Depending on the contrast this is typically something above 60 Hz.) Increasing the frame rate results in a reduced value of τ which also reduces the available contrast. The implication is that if higher line rates are to be used in the future, then monitors should have correspondingly shorter persistence phosphors, for equivalent performance.

The contrast is a function of $\lambda\tau$, so λ is inversely proportional to τ for constant contrast. Because of the nature of the human visual system this is not the case for a flicker-free field. Because the visual system has less sensitivity to flicker at higher frequencies, λ may be increased faster than τ is decreased. Figure 6 in Chapter 3 shows that at all spatial frequencies flicker visibility drops with increasing flicker frequency, for flicker frequencies greater than 10 Hz. The flicker contrast of a monitor displaying a static image is

$$\frac{1-e^{-\lambda\tau}}{1+e^{-\lambda\tau}}.$$

When $\lambda\tau$ is held constant as τ is increased, the flicker contrast and the available contrast are constant. When the visibility of flicker is constant, the actual contrast increases with increasing frequency, so that $e^{-\lambda\tau}$ decreases as τ decreases, and $\lambda\tau$ increases with decreasing τ. For example, a monitor refreshed at 30 Hz is flicker-free if $\lambda \leq 2s^{-1}$, while at 40 Hz, λ may be as high as $16s^{-1}$. When line rates approach the limits of human vision, the value of λ can be arbitrarily large, which means that the available contrast can become arbitrarily close to 1.

It is important to remember that the contrast is the local contrast, not the dynamic range. This is the maximum relative change from one frame to another, not the maximum change over an entire sequence of frames. Antialiased images or animated sequences rarely use the entire dynamic range in the interval of two adjacent pixels or frames, so this distinction is important.

Sometimes the phosphor decay cannot be described using a single exponential. The Electrohome model 38-D03101-60 monitor, for example, has two components in the red phosphor, a deep red shorter persistence one and an orange-red one with a longer persistence. In a static image the combination appears red, but when it is blanked, the fading image shifts colour toward orange. The one phosphor case, described above, and the two-phosphor case follow from the general p-phosphor case, which is derived next.

For each phosphor k and each frame j, define an intensity $I_{j,k}$. For each phosphor, define w_k, in analogy with w above. In addition, let α_k be the fraction of intensity attributable to the kth phosphor at the start of the first frame after a very long period of no excitation (a long blank period). Then $I_{j,k} = I_{j-1,k} w_k + J_j \alpha_k$, with J_j the intensity setting for the jth frame as above. This gives a recursive definition for $I_{j,k}$ in terms of J_j, J_{j-1}, etc.

$$\begin{aligned} I_{j,k} &= (I_{j-2,k} w_k + J_{j-1}\alpha_k) w_k + J_j \alpha_k \\ &= I_{j-2,k} w_k^2 + (J_{j-1} w_k + J_j)\alpha_k \end{aligned}$$

which expands to $I_{j,k} = \alpha_k \sum_{l=0}^{\infty} J_{j-l} w_k^l$. For sufficiently large n, $J_j = J_{j-ni}$.

Then $I_{j,k} = \alpha_k \sum_{i=0}^{\infty} \sum_{m=0}^{n} J_{j-ni-m} w_k^{ni+m}$

$$= \alpha_k \sum_{m=0}^{n} J_{j-m} \sum_{i=0}^{\infty} w_k^{ni+m} \quad .$$

The inner sum is a geometric series, so

$$I_{j,k} = \alpha_k \sum_{m=0}^{n} J_{j-m} \frac{w_k^m}{1-w_k^n}$$

$$= \frac{\alpha_k}{1-w_k^n} \sum_{m=0}^{n} J_{j-m} w_k^m \quad ,$$

which gives $I_j = \sum_{k=1}^{p} \frac{\alpha_k}{1-w_k^n} \sum_{m=0}^{n} J_{j-m} w_k^m$

$$= \sum_{m=0}^{n} J_{j-m} \sum_{k=1}^{p} \frac{\alpha_k w_k^m}{1-w_k^n}$$

in the case of p phosphors. From this form the available contrast may be calculated numerically, given the decay constants of the phosphors and the relative initial intensities attributable to them.

If the values of λ are not known, then values of J_j may be found manually by changing the settings of pixels behind a travelling square wave, in an attempt to make the trail disappear. This takes only one to two minutes for the single phosphor case, since only one pixel setting needs to be adjusted. For multiple phosphors, the manual method is quite tedious. For the Electrohome monitor with a two component red phosphor, the process takes between half an hour and an hour. Table 1 shows Ikonas look-up table values for a square wave of maximum contrast. In the experiment where they were found, the last value recorded before an edge is replicated, until the beginning of the next edge. The equivalent table for a single component phosphor would consist of constant values except for a single value at the leading and trailing edges. The inter-trial variance is partly a result of differences being hard to distinguish, but primarily results from frustration on the part of the observer. With a single component phosphor a very good result may be obtained, but with multiple components, the image is eventually good enough that the small improvements possible are not worth the time required. Several other factors contribute to the inter-trial variability of the data. One is that an error in any pixel affects all values beyond that value. It should also be remembered that the values tabulated are not intensity values, but the values used as voltages to drive the guns of a CRT. If these are converted to voltages using the inverse of gamma correction, the variation is less.

Table 1
Ikonas RDS 3000 lookup table values for a red travelling square wave on the Electrohome 38-D03101-60.

	Trial 1	Trial 2	Trial 3
Leading Edge	1023	1023	1023
	978	937	957
	958	915	937
	930	843	934
	911	878	894
	909	828	892
	900	833	884
	900	860	859
	885	832	864
Trailing Edge	0	0	0
	356	272	431
	566	482	526
	628	572	580
	680	613	601
	668	614	612
	718	636	645
	692	645	640
	713	654	644

8.1.3. Frequency space minimization

For comparison with the direct method, we consider the result of using the frequency space method with an exponential pixel decay function. From Section 5.3, the problem is to minimize $||W_n(z_n - z_n')||$, where

$$z_n' = \frac{2\sqrt{2\pi}}{N\tau} \sum_{j=1}^{N} a_j \mathcal{F}[B(t - j\tau), \omega_n] = [\mathbf{Ma}]_n$$

with $B(t) = \lambda \tau e^{-\lambda t}$. z_n is the nth complex Fourier series coefficient of the desired output function, and a_j is the intensity value of the jth frame. Section 5.3 shows that this is a matrix-vector problem. The matrix depends on the device, the visual system's sensitivity function and the number of frequencies being included, and the right hand side vector is derived from the Fourier series coefficients of the desired image. The solution gives the pixel intensities. From Theorem 2, and the Fourier transform of the exponential function,

$$\mathcal{F}[B(t-j\tau),\omega_n]=e^{i\omega_n j\tau}\mathcal{F}[B(t),\omega_n]=\lambda\tau\frac{e^{i\omega_n j\tau}}{\sqrt{2\pi}}\frac{(\lambda+i\omega_n)}{(\lambda^2+\omega_n^2)}\ .$$

Expanding the complex exponential and substituting in the expression for z_n' gives

$$z_n'=\frac{2\lambda}{N}\sum_{j=1}^{N}a_j\frac{\lambda\cos\omega_n j\tau-\omega_n\sin\omega_n j\tau+i(\omega_n\cos\omega_n j\tau+\lambda\sin\omega_n j\tau)}{\lambda^2+\omega_n^2}\ .$$

For minimizing in the l_1 and l_∞ norms (Section 5.1), let

$$M_{0,j}=\frac{1}{N}$$

$$M_{2n-1,j}=2\frac{\lambda}{N}\frac{\lambda\cos\omega_n j\tau-\omega_n\sin\omega_n j\tau}{\lambda^2+\omega_n^2}$$

$$M_{2n,j}=2\frac{\lambda}{N}\frac{\omega_n\cos\omega_n j\tau+\lambda\sin\omega_n j\tau}{\lambda^2+\omega_n^2}.$$

The first row of M and even rows thereafter are the real coefficients $a_n'=\mathrm{Re}\,z_n'$, and odd rows after the first are the imaginary parts $b_n'=\mathrm{Im}\,z_n'$. If the right hand side vector is represented in the same way, then such a matrix may be passed directly to an l_1 or l_∞ linear programming package, possibly accompanied by constraints on the values of the result, which is the vector of pixel intensities.

A simple test case is the square wave. Note that representing a square wave by a finite number of evenly spaced frequencies is a band-limited approximation. It still interpolates the square wave at the sample points corresponding to a regular sampling at twice the highest frequency, according to Whittaker's theorem (see the Appendix). This function is better than a square wave for the purposes of this example, because it is band-limited, as required by our assumptions for the input.

If the square wave is symmetric about the origin, then $b_n\equiv 0$. Let one period of the square wave be defined as

$m+a$, $\quad |t|<t_s T$

$m-a$, $\quad t_s T<|t|<T$.

Such a wave has period $2T$, amplitude a, is centred vertically at m, and it is high a fraction t_s of each period.

The Fourier series coefficients are

$$a_0 = 2a\,t_s + m - a \quad ,$$

$$a_n = \frac{4a \sin n\pi t_s}{n\pi} .$$

Solving in the l_2 norm involves forming a new matrix-vector problem, in which the Fourier series coefficients of the pixel basis are used in the new matrix and right hand side vector. From Section 5.3, the definitions of this matrix and vector are

$$c_{j,k} = \frac{2\sqrt{2\pi}}{T} \sum_n W_n^2 \cos(\omega_n (j-k)\tau)\, \mathcal{F}[B(t),\omega_n]^* \mathcal{F}[B(t),\omega_n]$$

(which is symmetric circulant), and

$$b_k = \sum_n W_n^2 (\mathcal{F}[B(t-k\tau),\omega_n]^* z_n + z_n^* \mathcal{F}[B(t-k\tau),\omega_n]) \quad .$$

For the exponential decay,

$$\mathcal{F}[B(t),\omega_n]^* \mathcal{F}[B(t),\omega_n] = \lambda^2 \frac{\lambda^2+\omega_n^2}{2\pi(\lambda^2+\omega_n^2)^2} = \frac{\lambda^2}{2\pi(\lambda^2+\omega_n^2)} \quad ,$$

so that an element of the matrix is

$$c_{i,j} = \frac{2\lambda^2}{T\sqrt{2\pi}} \sum_n W_n^2 \frac{\cos(\omega_n (j-k)\tau)}{(\lambda^2+\omega_n^2)} \quad .$$

Solving $\mathbf{Ca} = \mathbf{b}$ gives the vector of intensities which minimizes the error under the l_2 norm.

On the assumption that $W_n \equiv 1$ for all frequencies less than a cutoff frequency (treating the visual system as a perfect low-pass filter), the sum becomes

$$\sum_{n=0}^{cutoff} \frac{\cos(\omega_n (j-k)\tau)}{(\lambda^2+\omega_n^2)} \quad .$$

If *cutoff* is very large, the sum is expensive to compute directly. For large enough values of *cutoff*, a closed form for the infinite sum may be substituted for the finite sum, which puts a limit on the number of terms that might ever need to be computed. $\sum_{n=1}^{\infty} \frac{\lambda^2 \cos\omega_n (j-k)\tau}{\lambda^2+\omega_n^2}$ has the form $a^2 \sum_{n=1}^{\infty} \frac{\cos n x}{a^2+n^2}$, where $x = 2\frac{\pi}{T}(l-k)\,\tau = 2\frac{\pi}{N}(l-k)$ and $a = \frac{\lambda T}{2\pi}$.

The closed form for this sum [Gradshteyn1965] is $\frac{\pi}{2a} \frac{\cosh a(\pi - x)}{\sinh a\pi} - \frac{1}{2\pi}$,

$$\text{so that } M_{l,k} = \frac{1}{N^2} + \frac{4}{N^2}\left[\frac{\lambda T}{4}\frac{\cosh\frac{\lambda T}{2}(1-\frac{2}{N}(l-k))}{\sinh\frac{\lambda T}{2}} - \frac{1}{2}\right]$$

$$= \frac{1}{N^2}\left[\lambda T\frac{\cosh\frac{\lambda T}{2}(1-\frac{2}{N}(l-k))}{\sinh\frac{\lambda T}{2}} - 1\right].$$

In practice, the number of terms required is usually small enough that the closed form solution is not a good approximation to the finite sum. It is useful only for very long persistence phosphors, for which $\lambda\tau$ is exceptionally small. Typical cutoff values are less than 75% of the way to being large enough that the closed form gives a good approximation to the finite sum.

8.1.4. Comparative Results

In this section a number of the tradeoffs are contrasted using a square wave as an example. The examples were computed in two different ways: a 64 frame sequence was generated for display (using the methods in Chapter 7); the tables included in this section show a similar 16 frame sequence which illustrates the important points equally well. In both cases the sequence has a period of one second, and $\lambda\tau$ is the same in both.

Several considerations have a qualitative effect on the solutions generated. When the number of sample points is the same as the number of pixels, and there are no constraints, the direct method finds an exact match, and the choice of norm is irrelevant. The examples in this section show the effects of the choice of norm in the presence of constraints, for both the direct and the frequency space methods. When the number of sample points is greater than the number of pixels, aliasing can result, unless the sampled image has been filtered prior to sampling. This is discussed further in Section 8.2, using Gaussian pixels as examples.

Table 2 shows the pixel values generated by constrained minimization in which small negative values would occur in the absence of constraints. (The length of the residual is the same as the matrix width; the length of the result vector is the same as the height). The results of l_1 and l_2 are barely distinguishable, while the result of l_∞ minimization is quite different. Once the worst error is as small as possible, none of the other errors affect the l_∞ norm, and the result, while not as good as it could be, is l_∞ optimal. Of particular note are the largest elements of each residual. These are emboldened in the l_1 and l_2 residual vectors. The elements of the l_1 and l_2 residual

Table 2: Comparison of different norms - Direct Method					
Values			Residuals		
l_1	l_2	l_∞	l_1	l_2	l_∞
0.0777	0.0777	0.0137	–3.469e–18	1.735e–18	–0.0824
0.0777	0.0777	0.0137	–3.469e–18	1.735e–18	–0.0824
0.0777	0.0777	0.0137	–3.469e–18	1.735e–18	–0.0824
0.0777	0.0777	0.0137	–3.469e–18	1.735e–18	–0.0824
0.0777	0.0777	0.0137	–3.469e–18	1.735e–18	–0.0824
0.0777	0.0777	0.0085	–3.469e–18	1.735e–18	–0.0824
0.0526	0.0563	0.0000	–3.469e–18	1.735e–18	–0.0593
0.0117	0.00000	0.0000	**0.1125**	**0.0960**	0.0824
0.6992	0.6778	0.6351	5.551e–17	–0.0214	–0.0824
0.6992	0.6992	0.6351	5.551e–17	2.776e–17	–0.0824
0.6992	0.6992	0.6351	5.551e–17	2.776e–17	–0.0824
0.6992	0.6992	0.6351	5.551e–17	2.776e–17	–0.0824
0.6992	0.6992	0.6351	5.551e–17	2.776e–17	–0.0824
0.6992	0.6992	0.6351	5.551e–17	2.776e–17	–0.0824
0.6992	0.6992	0.6351	5.551e–17	2.776e–17	–0.0824
0.8777	0.8777	0.8137	5.551e–17	2.776e–17	–0.0824
			5.551e–17	2.776e–17	–0.0824

vectors are practically identical (and zero) except for the largest elements. In the l_2 case an increased error in the second largest element is traded for a decreased error in the largest. The visual system is able to detect discrepancies larger than the *Weber fraction*, about 1% [Kaufman1974], so only the large errors in the l_1 and l_2 norms are visible. All the errors in the l_∞ solution are visible.

Minimizing the point of greatest error while ignoring all others gives a poorer result than minimizing one of the other norms in the direct minimization approach. In the frequency space approach, there is less reason to choose the l_1 norm over the l_∞ norm. Given that the visual system detects different frequencies separately, the right perceptual criterion might easily be minimization of the greatest error in the frequency domain. The errors at all other frequencies might be almost as great, but if the greatest error is small enough to be invisible, there would be no visible artifact. This analysis assumes a perfectly linear visual system, with equal sensitivities to all frequencies.

Table 3 shows the results of the three norms, under equivalent conditions to Table 2.

Table 3: Comparison of different norms – Frequency Space

Values				Residuals		
l_1	l_2 (constrained)	l_2 (unconstrained)	l_∞	l_1	l_2 (unconstrained)	l_∞
0.9839	0.9538	0.9555	0.9347	2.776e−17	1.926e−34	0.0082
0.8912	0.8794	0.8745	0.8788	0.0000	0.0000	−0.0082
0.9311	0.9751	0.9974	0.9929	0.0000	0.0001	0.0082
0.0000	0.0000	−0.0600	0.0000	0.0000	0.0000	0.0082
0.0000	0.0000	0.0026	0.0000	0.0000	0.0000	0.0082
0.0839	0.1233	0.1255	0.1201	0.0000	0.0001	0.0082
0.0716	0.0445	0.0445	0.0532	0.0000	0.0000	−0.0082
0.1310	0.1124	0.1129	0.1740	0.0000	0.0000	0.0082
0.0161	0.0462	0.0445	0.0441	0.0000	0.0000	0.0057
0.1088	0.1206	0.1255	0.1454	−0.0296	0.0001	−0.0082
0.0689	0.0249	0.0026	0.0200	0.0000	0.0000	0.0082
1.0000	1.0000	1.0600	1.0000	0.0000	0.0000	0.0075
1.0000	1.0000	0.9974	1.0000	0.0000	0.0000	0.0082
0.9161	0.8767	0.8745	0.8743	0.0116	0.0001	0.0082
0.9284	0.9555	0.9555	0.9363	0.0000	0.0000	−0.0009
0.8690	0.8876	0.8871	0.9579	0.0000	0.0000	0.0039
				0.0000	0.0000	0.0082

There are two things worthy of note in this table. One is the relative quality of the l_2 and l_∞ solutions, the other is the nature of the l_∞ solution.

The l_2 solution is better in the l_∞ sense than the l_∞ solution. Since different algorithms were used to find the solutions, this discrepancy is likely the result of different round-off errors.

The l_∞ residual also tends to change sign frequently. The result is that while the individual errors are smaller, they happen to add up in such a way as to cause the solution to oscillate. Presumably the phase is aligned so as to make the error particularly bad. This problem is easier to see in the solution of a larger system, such as $N = 64$. Ringing at the edge of the square wave is well below the precision of current display devices after a few oscillations, in the l_2 solution. Contrarily, it continues significantly further for l_∞. In addition to the slightly better result numerically, the l_2 norm solution may be found much more quickly than the l_1.

Oscillations in the solution are undesirable, but only in the frequency space version is the solution based on a band-limited square wave. A non-band-limited square wave contains a discontinuity, which causes Gibb's phenomenon, oscillations in

the solution at points close to a discontinuity. When the values found by frequency space minimization are used for display, oscillations are not visible on the screen. The overall appearance is one of very slight edge enhancement over the solution by the direct method. By the direct method, the moving square wave looks very much like a moving square wave. In fact it looks so good that it seems like no improvement is possible, until the solution of the frequency method is displayed. The solution of the frequency space method looks just slightly better, in the sense that it looks more as a square wave should. The oscillations are not visible, and the edge looks even sharper.

From the examples discussed in this section, it appears that the l_2 norm gives results which are either superior to, or visually indistinguishable from the results of minimization under the other norms. The results of the frequency space method are slightly better than the results of direct solution, but it is unlikely that the difference will be noticeable in any but contrived images, since it is small. The l_∞ norm is not likely to be useful for work of this sort, even in the frequency space method.

8.2. Gaussian Pixels

The Gaussian basis function is

$$B(x) = \frac{\delta}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2} .$$

It is normalized to provide unit integrated intensity in unit distance with pixels spaced δ apart. If the pixel values repeat every N as before, then the effect of the ith pixel value at $x = j\delta$ is

$$b_{ij} = \sum_{p=-\infty}^{\infty} B((pN + i - j)\delta)$$

$$= \sum_{p=-\infty}^{\infty} \frac{\delta}{\sigma\sqrt{2\pi}} e^{-((pN+i-j)\delta)^2/2\sigma^2} .$$

This is analogous to the matrix **B** used for the exponential decay. Because no closed form exists for the sum of Gaussians, a finite sum is used to calculate matrix values. The elements of the series being summed drop quickly with increasing distance from the highest, so the number of elements needed is small. Where the exponent is less than -11, the contribution is less than one part in 10^5 as great as at the centre of the pixel. For $\sigma = 0.5\delta$, which is greater than that for a properly focussed CRT, $(pN + i - j)$ must have magnitude less than 6 for any contribution large enough to be seen to result from that term and all others further away from the pixel centre. This means that for this value of σ, not more than 11 terms are needed to approximate the infinite sum.

If the sampling rate is not the same as the pixel rate, the matrix is pseudo-circulant, not circulant. With the pixel rate at τ', its elements are

$$b_{ij} = \sum_{p=-\infty}^{\infty} \frac{\delta}{\sigma\sqrt{2\pi}} e^{-((pN+i)\delta - j\delta)^2/2\sigma^2} .$$

The quality of images produced with δ functions simulating Gaussians improves with the number of δ functions per Gaussian being simulated. Going the other way, using many more samples than Gaussian pixels potentially makes matters worse, rather than better. When there are more samples than Gaussians, high frequency information may be introduced, which cannot be well approximated by the sum of Gaussians. Since the sum of errors is minimized the solution contains both high and low frequency errors. Figure 22 shows the effect of allowing high frequency information to be present in the input in this way. Note particularly the lack of symmetry at the edges of the square wave in b) and c). This is a result of aliasing.

One result of introducing more high frequency information is an increase in the maximum value of the derivative of the image function, causing the range of the image function to be greater. This increase is visible in the pixel values as the amount of high frequency information is increased. Because the sampling grid is finer, more overshoot is required to interpolate close to the discontinuities. The lack of symmetry in the square wave is a result of aliasing.

When the image is filtered prior to sampling, and the high frequency information is removed, the problems associated with aliasing cannot occur. In Figure 23 the finite Fourier series of a square wave with eight frequencies is sampled in 16, 18 or 20 places to provide the input. The same matrices were used as in Figure 22. The results are nearly independent of the sampling rate, and the image has the proper symmetry.

8.3. The Solution as a Filter

The multiplication of a vector by a circulant or pseudo-circulant matrix is mathematically equivalent to convolution, which is equivalent to filtering. The filters for the exponential and Gaussian cases are given in Table 4. The exponential filter has only two non-negligible entries. This is in agreement with the fact that only one frame needs to be corrected in a pure exponential decay model. The Gaussian filter has seven entries large enough to make a significant difference to the result (an error of less than one part in 2000 is well below the 1% Weber fraction). Both of these filters are sufficiently narrow to be practical for use in computer graphics.

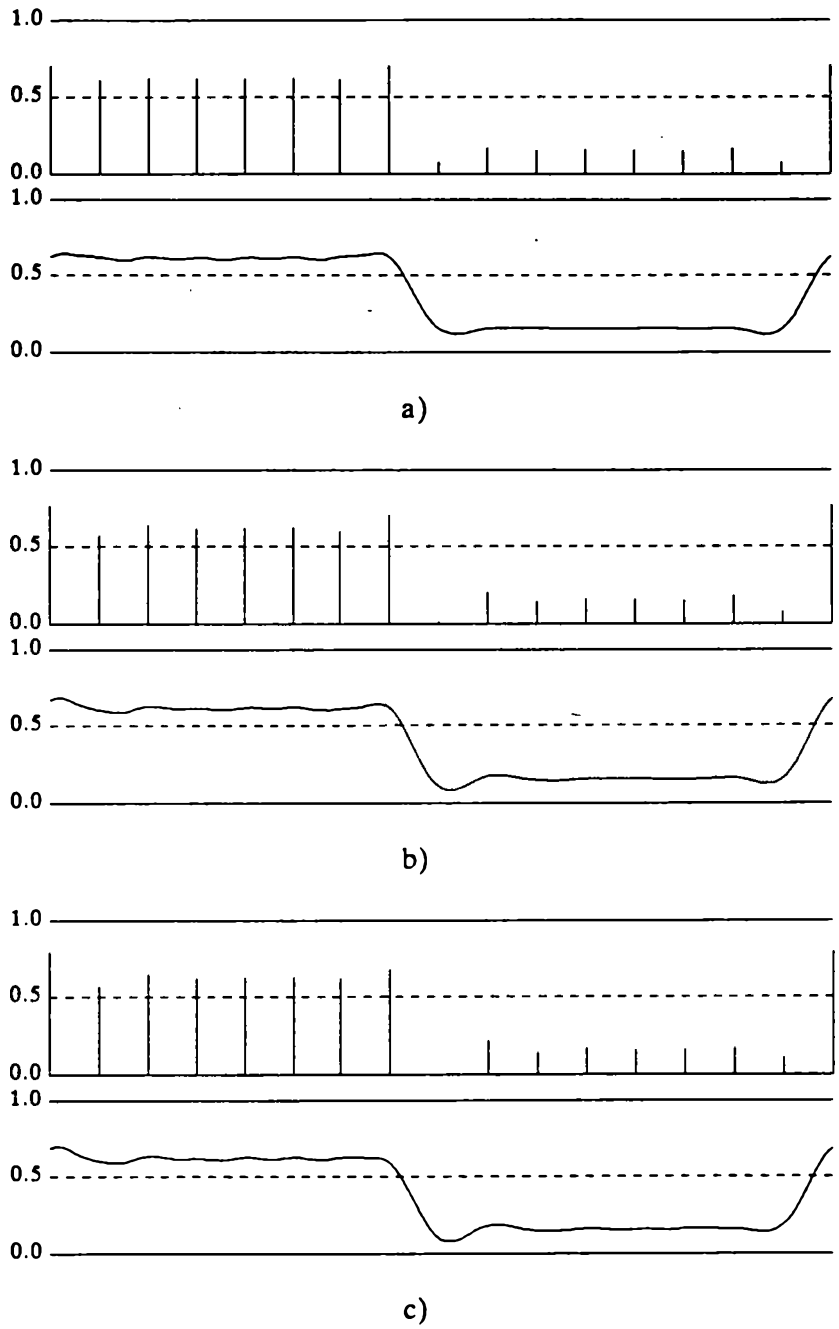

Figure 22: A square wave corrected for a Gaussian pixel. $\sigma = .51\delta$. 16 pixels per period. The number of samples in the input is a) 16, b) 20, c) 24. The width of the square wave is half the period in each case, but as the number of samples increases, the transition occurs on a region more and more narrowly specified.

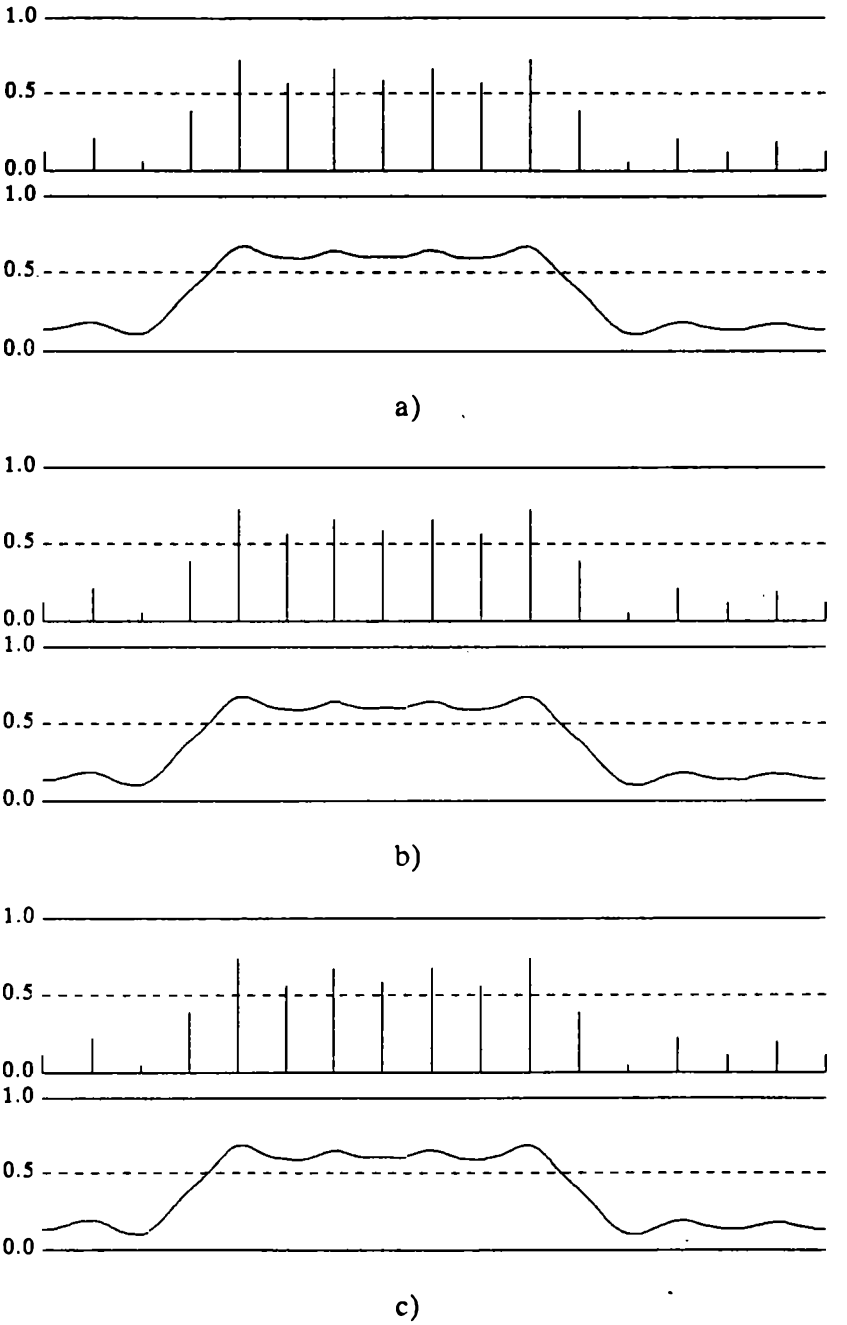

Figure 23: A square wave corrected for a Gaussian pixel. Parameters as in Figure 22, except that the square wave is constructed as a Fourier sum of 8 sinusoids, so its highest frequency is at the Nyquist rate for the display.

Table 4: Matrix and Filter Values			
Exponential ($\lambda\tau=1$)		Gaussian ($\sigma=.51\delta$)	
Matrix	Filter	Matrix	Filter
1.00000	1.00000	1.00000	1.04495
0.36788	–0.36788	0.14516	–0.15488
0.13534	0.00000	0.00044	0.02249
0.04979	0.00000	0.00000	–0.00327
0.01832	0.00000	0.00000	0.00047
0.00674	0.00000	0.00000	0.00007
0.00248	0.00000	0.00000	0.00007
0.00091	0.00000	0.00000	0.00047
0.00034	0.00000	0.00000	–0.00327
0.00012	0.00000	0.00044	0.02249
0.00005	0.00000	0.14516	–0.15488

The existence and narrow width of the filters are significant because filtering is computationally practical. To correct still images, for each pixel in an uncorrected image, compute a new value using a weighted average of the pixels in its neighbourhood. If the filter values appropriate for a given pixel basis are used as weights, the resulting image is corrected for the pixel shape. In a similar way frames can be corrected for the effect of preceding frames. Because of the particularly simple filter for the exponential basis, this amounts to calculating a frame of correction values from the first frame in the sequence, and then subtracting it from the next frame. The resulting frame is the second frame in the filtered sequence, and from it the next frame of correction values is calculated. In a single pass through an entire animated sequence, consisting of the simple loop

```
for i from 1 to number of frames
  multiply frame i by filter value giving correction frame.
  replace frame i+1 with frame i+1 minus correction frame.
end
```

the effect of exponential decay can be corrected.

If the pixel basis is other than those for which the filter has been shown above, it can be readily calculated by building the circulant matrix which appears in the formulation of the l_2 solution, and inverting it. As shown in Section 5.3.3, the inverse matrix has as its values samples of the filter at the points for which it is needed.

8.4. Summary

Two one dimensional pixel profiles are considered in this chapter: the exponential decay, and the Gaussian. The exponential has a simple closed form solution, while the Gaussian has to be solved in matrix form. As the effective pixel width increases, the available contrast decreases, because it is not possible to change the intensity as much from one pixel to the next. The image should always be free of coherent high frequency information before sampling. One way of ensuring this is to use the frequency space method, another is to prefilter the image.

Of the three norms considered, the l_2 norm is the least expensive, and for the two pixel basis functions considered in this chapter, gives results which are never worse than those given by the other norms. The l_∞ norm is a poor choice, even in the frequency domain.

The method works for resizing of images as well as compensation for device pixels on devices at the same resolution as the image. In resizing, if the device has fewer pixels than the number of samples, then the image should be filtered to prevent information above the Nyquist limit for the device.

Two Dimensional Image Construction

An image specified in two dimensions can be a static picture or an animated sequence of one dimensional spatial images. If the pixel profile is separable a solution can be found using the direct method in two passes, as shown in Section 5.5. To the extent that the response function of the human visual system is separable, the frequency space approach can use the efficient two pass technique as well. The two pass technique solves the problem in each dimension independently and for each dimension finds a solution in the same manner as discussed in Chapter 8. This chapter explores the effects of using a particular separable response function, in contrast to using a good (non-separable) approximation to the actual response function.

On a CRT the y-profile of a pixel and the x-profile are much the same. For the purposes of this thesis they are approximated as Gaussians, so the two dimensional pixel profile is radially symmetric. Both the two dimensional spatial and the spatio-temporal pixel basis functions are separable. Other than a small correction, the human visual system's spatial response is separable into the product of its responses to horizontal and vertical variations. Because both the visual system and the pixel shape are separable for this case, it can be solved in two passes. A two pass technique is available regardless of whether the solution is found in frequency space or directly, for any pixel basis which is separable into a product of two one dimensional functions.

When the second dimension is time, the response of the human visual system is no longer separable. If the human visual system is not taken into account, then any separable pixel basis can be used in a two pass solution with one dimension being time. If the human visual system is taken into account, the two pass solution does not apply to the frequency space solution, with a non-separable sensitivity function for the human visual system. Some approximations to the response of the visual system are separable. Their use gives different results from the results of an accurate model, but requires much less computation. This chapter addresses the question of how important the use of the more accurate non-separable model is, compared to the separable approximations. If the differences between the results are not visible, then

the extra computation is not worthwhile.

If no compensation is made for the device, the image resulting can be considerably less sharp than possible. Figure 24 shows the result of specifying the intensities for a moving square wave as if no device compensation were necessary. The curves show the spatial intensity of each frame. The intensity is "calculated" by displaying it on a CRT. Each of the spatial profiles is considerably less sharp than the profiles in the last chapter, which are calculated with one dimensional compensation. The image has no sharp edges, as a result of the blurring inherent in the pixel basis. Very little high frequency information has been retained.

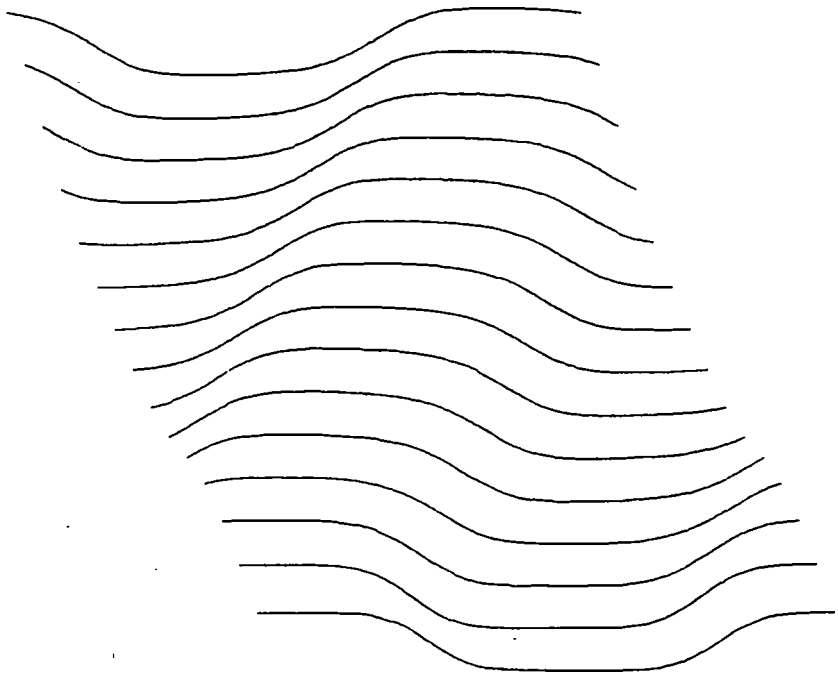

Figure 24: A moving square wave constructed as if no device compensation were necessary, and then displayed on a CRT. Each frame is shown displaced slightly to the left and up from the previous one (position increases to the right, time increases upward). The square wave is moving from left to right, and moves across the entire display every 16 time intervals. Note that each frame is an exact translation of the previous.

Each of the methods explored in the last chapter can be applied in two dimensions. In this chapter the example device is the CRT, with Gaussian pixels spatially and exponential pixels temporally. In Chapter 8, it is shown that for these two pixel bases, minimizing the l_2 norm produces the same results as minimizing the l_1 norm, but in less time, and they both produce better results than the l_∞ norm. For this reason, only the l_2 norm is considered in this chapter. For the direct method, it is

shown in Section 5.5 that two dimensional solutions in the l_2 norm may be obtained by two passes of the one dimensional technique. (The direct method always neglects the nature of the human visual system). For band-limited input images, the direct method gives almost the same results as the frequency space technique if all frequencies are weighted equally. This follows from the definition of Fourier series, and is demonstrated in Section 8.1.4 to hold for the case of exponential decay. For equal weighting of different frequencies with a separable sensitivity function, frequency space minimization can be performed in two passes, as shown at the end of Section 5.5.

The interesting question remaining is how much difference does it make whether a realistic sensitivity function is used, or whether a separable approximation is used instead. The way in which this is addressed is to find the solution both with a separable approximation, and with the surface constructed from Robson's data in Chapter 3, and compare the results. The mathematics has all been covered in Chapters 5 and 8, so only results are presented in this chapter.

The separable approximation used in this chapter weights all frequencies equally up to the Nyquist frequency for the display, and gives zero weight to all others. A uniform weighting with a higher cutoff value has serious aliasing artifacts, just as it does in one dimension. The number of frequencies with non-zero weight is one greater than the number of pixels, ($N/2$ negative, $N/2$ positive, and 0), so the problem is slightly overdetermined. The results are almost exactly the same as if the problem were solved using the direct method (differences are around one part in 10^{-8}). For this reason only results for the frequency space method are presented in this chapter.

Figure 25 shows the results of weighting frequencies equally. Note that there is some ripple in the waveform, particularly in the trough. The data in Figure 25 is less meaningful on its own than the images viewed on the screen, displayed using the methods of Chapter 7. When a single frame is viewed in isolation, the ripple is visible. When the frames are set in motion, the ripple disappears. Thus ripple at this combination of spatial and temporal frequency is not visible. Referring back to the contour diagram of Koonderink's spatio-temporal response function (Figure 8 in Chapter 3), the top right corners of the contours are rounded. At 0 Hz, the spatial ripple is above threshold, but at the temporal frequencies involved in the moving display, it is not. Since the visual response function is not used in computing the waveform, the invisibility of ripple is not so much by design as by chance.

If frequencies up to the Nyquist limit are weighted with sensitivities corresponding to the spatiotemporal surface graph in Chapter 3, then the results are scarcely distinguishable from those shown in Figure 25. The differences are

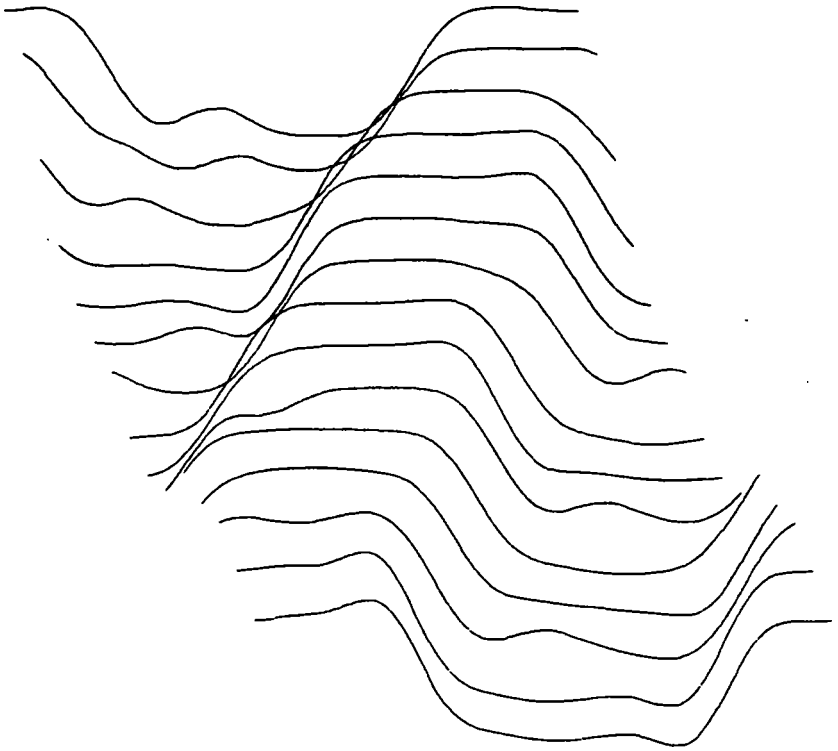

Figure 25: A travelling square wave calculated with all frequencies up to the Nyquist rate weighted equally. The picture for weighting according to the spatiotemporal response function is identical to this resolution, as long as only frequencies up to the Nyquist rate are given non-zero weights. Frames are no longer translations of each other. The steepness of the edge has improved significantly, at the cost of introducing some ripple.

approximately one part in 10^{15}, in pixel values in the range of 0..1, which is just enough that roundoff error is not the only source of the difference, but little enough that the difference is well below the precision of any device used to display the image. As frequencies beyond the Nyquist limit are added, the results slowly diverge from those in Figure 25. When frequencies up to the limit of the human visual system are considered, the image appears as in Figure 26.

In both figures the square wave is band-limited by using the Fourier expansion up to the Nyquist limit, and zeroes beyond. The difference between the two is that in Figure 26, errors in frequencies beyond the Nyquist limit (non-zero amplitudes) are possible. These result in some aliasing, which is traded for reduced errors in frequencies below the Nyquist limit, which result from the pixel shape. The artifacts in Figure 25 are not visible. This means that the differences between Figures 25 and 26 are not large enough to be visible. For the particular case of the CRT, it appears

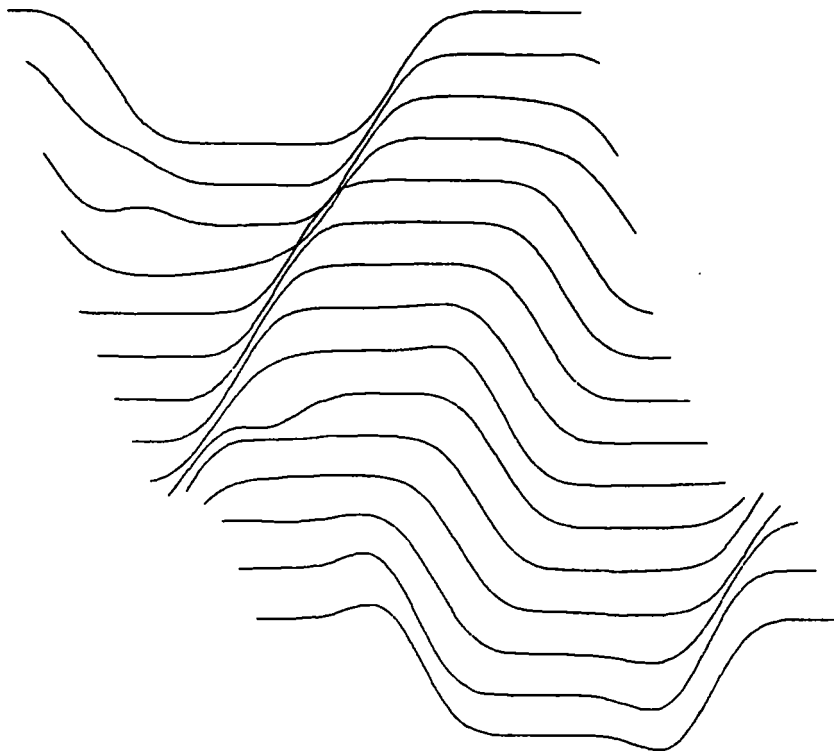

Figure 26: The same wave as in Figure 25, but constructed with the full response of the human visual system taken into account. The steepness of the edges in Figure 25 is hardly changed, but the ripple is nearly eliminated.

that the use of weights based on the human visual system's responses does not add any quality, in a visual sense. While the differences between the data of Figures 25 and 26 are not visible when the data is displayed as an animated image, Figure 24 is much different. The animated image corresponding to Figure 24 has a less sharp edge to the square wave, but more noticeably, a trail of decaying phosphor behind the trailing edge.

While the elimination of pixel structure artifacts in Figure 26 is quite effective. the artifacts remaining in Figure 25 are quite small. The visibility of such artifacts is sufficiently small as to be unimportant. The two pass technique can be used to find the values used for Figure 25; it cannot be used to find the values used for Figure 26. The difference, while it shows up on the graph, is invisible in practice.

The figures show small images: 16 frames with 16 pixels per frame. There were two reasons for this. One reason was presentation. Too many frames or too many pixels would make it difficult to see the information contained in the figures. The other reason was computation time. To compute the solution for Figure 26 took

overnight on a VAX 8600, and the algorithm is quadratic in the number of pixels in each dimension. The two pass technique takes time $O(nm \log n \log m)$ where n and m are the numbers of pixels in each dimension. Because the two pass technique lends itself to implementation as a filtering method, the $O(nm \log n \log m)$ time can be off-line preprocessing cost, in which case the time required for an image is proportional to the total number of pixels, or $O(mn)$. Even without the efficiency of filtering, computations such as the one required to compute the data for Figure 25 take minutes rather than hours to find a solution. Computations that size are sufficient to define a filter, which then can be used on practical problems, yielding solutions at a rate of several frames per minute on a VAX 8600.

The results of the two techniques are similar, but not indistinguishable, at least numerically. For some displays they are likely to be easily distinguished visually. The computation times are not at all alike, the single pass version being too slow for any practical use. For these reasons, the two pass technique with either equal weights at all frequencies up to the Nyquist limit, or a separable approximation to the spatiotemporal response function is recommended.

Discussion

Aliasing artifacts result from attempting to display high frequency information on a device with insufficient resolution. Examples of aliasing artifacts are Moiré patterns in regular textures, and jaggedness of near-horizontal and near-vertical lines. Pixel structure artifacts are the result of ignoring a specific property of the real device used to display any raster image: the device is made up of pixels, which have an influence over a region of the display. Examples of pixel structure artifacts are loss of sharp edges due to a Gaussian spread, and fading trails behind moving objects due to an exponential decay. Under the right circumstances, both of them can be reduced or avoided entirely. This thesis addresses the removal of pixel structure artifacts. The methods used to remove them can be used to remove aliasing artifacts at the same time, and thus generalize filtering techniques found in the literature.

Several methods were explored. Of these, direct minimization of discrepancies between the ideal image at sample positions and the device image at the same positions is representative of methods which include comparisons with filtered versions of the ideal image. Distinct from these methods is minimization of errors in frequency space. Three norms are used as measures of the size of the error in the results. Of these, the l_∞ norm produces equivalent or inferior results to those produced by the l_2 and l_1 norms, which produce results which are nearly identical to each other. Finding minima using the l_2 norm generally involves inverting circulant matrices, which can be done very efficiently using the Fast Fourier Transform.

Two particular pixel shapes are considered as examples. They are characteristic of the temporal and spatial pixel profiles of an idealized CRT. For these two shapes, the results of frequency space methods are the same as the results of the direct method, to within device precision. Other pixel shapes may require use of the frequency space method to give the best results.

For either method (frequency space or direct), some images cannot be displayed without needing intensities outside of the range of the device. Constrained optimization is one way of dealing with this. This is a local solution, in that corrections are made to a small region of the image, or to a small number of

frequencies, to prevent the frame buffer values from being outside of the acceptable range. The results of constrained minimization depend on the extent to which constraints are needed. If the intensity values required without constraints violate the constraints significantly, then a poor quality image results. If not, then the image produced with constraints has small errors introduced by the constraints. The other way of dealing with images that need intensities outside of the device range is to reduce the dynamic range of the image until the contrast is reproducable by the device. This is a global change, in that the entire image is changed, but no particular part of the image is singled out. Just as for constrained optimization, an image which can almost be displayed without needing values outside of range is only slightly degraded. One which needs significant correction results in a poor quality image. Using the contrast reduction approach allows the minimization to be carried out in an unconstrained form, which results in considerable reductions in computation time.

When minimization is carried out in the l_2 norm with no change in size, the inverse of the matrix for the calculation defines a filter, which can be used for space- and time-efficient solution for any number of images on the same device. For the two pixel basis functions considered in this thesis (Gaussian and exponential), the filter is only a few pixels wide, making the filtering operation fast. This technique should result in a considerable improvement in the quality of images displayed on CRTs. For increases in resolution, the matrix multiplied by the input vector (a pseudo-inverse) is pseudo-circulant; it also defines a filter (the same filter as for no change in size), sampled at all the positions needed for enlarging by digital filtering. It is a simple matter to construct the filter from the matrix. The width of this filter is the product of the width of a filter for no change in size and the scale factor for resizing, which makes this as fast as filtering without resizing an image the size of the enlargement. For reducing image size, a band-limiting filter should be used to prevent aliasing.

With the methods in this thesis some pixel structure artifacts can be avoided. On very high resolution devices, pixels which induce fewer artifacts can be simulated. There remain ample opportunities for further work, as many questions remain unanswered. One is optimal pixel shape. The measurements presented in Section 7.5 provide a partial answer. The best Gaussian pixel spread is directly related to the contrast threshold of the human visual system. It is a compromise between giving the best contrast between pixels which are alternating between full on and off, and giving a good flat, flicker-free field in an image with all pixels full on. For Gaussian pixels, the flat field has low amplitude oscillations which contain nearly no information above the fundamental frequency. The minimum Gaussian spread for the pixels is readily derived from the contrast threshold of the human visual system. If the pixel is broader than the minimum, then the device is not able to change the intensity

between pixels as rapidly as possible, and so any image displayed is more blurred than necessary. A good measure of the device's ability to change its intensity from one pixel to another is the contrast of an image with pixels alternating between on and off. In such an image, pixels with too broad a spread reduce the contrast. If these were the only two considerations (quality of flicker-free flat field and contrast of alternating pixels) the ideal pixel would be a flat square pixel, i.e., one which had no influence outside of the region between pixel midpoints, and equal influence everywhere within. One other criterion is important: there should be a minimum of visible information above the fundamental in an image of alternating scanlines. Since an antialiased image contains no information above the fundamental frequency in an image of alternating scanlines, any information in such an image is introduced by the pixel shape and results in artifacts in all images produced on the device. For this reason the Gaussian is a better choice than the flat square pixel response. Finding the best possible spatiotemporal pixel shape is still an open problem. It may be a three dimensional Gaussian – the lack of harmonics in the flat field constructed of Gaussians is suggestive – but that remains to be proven. Whichever pixel shape is best, it can be simulated on a very high resolution device using the methods in Chapter 8.

Solutions to the general problem of selecting pixel values for an image which is known only from a sampling are given only for regularly spaced sample points. Because of the effectiveness of stochastic sampling as a method of removing aliasing, it is very much worth trying to find an efficient method of finding a good set of device pixel values for an image which is known from a sampling with irregular intersample spacing. A global method is easy to state, but much too slow to be of any practical interest in the foreseeable future: the error at the sample points may be minimized in the least squares sense. This is conceptually similar to the l_2 minimization described in Chapter 6. This would involve a system of as many equations as there are sample points, in as many unknowns as there are pixels. For a 512×512 image, which is toward the low end of devices in use today, there are 2^{18} unknowns and a similar number of equations. For this reason such a method is impractical. Since pixels only have an influence over a small region of the screen, the matrix contains a narrow band in which values are non-negligible, so that banded matrix techniques could be used to speed up the process, but the size of the matrix is still such as to keep the solution process slow. The narrow influence of pixels suggests the existence of a local solution technique, which is probably close in expense to the cost of methods currently in use to combine the values of multiple samples to find the values assigned to a pixel.

The frequency space method is instructive in the similarity of its results to those of the direct method for the pixel shapes explored in this thesis. It would be interesting to see whether this is the case for other reasonable pixel shapes.

Appendix

Fourier analysis has already been introduced as a means of describing a signal by the phases and amplitudes associated with a set of sinusoids at various frequencies. A more complete definition is given here, including theorems which indicate how to obtain the Fourier expansion of a known function.

Fourier series provide a means of expressing many functions in terms of pure sine and cosine functions.

THEOREM A.1: *Any function $f(\xi)$, defined over a finite interval $0..\Delta$ or periodic, with period Δ, and having finitely many finite discontinuities in the interval, may be represented uniquely as the sum*

$$f(\xi) = a_0 + \sum_{n=1}^{\infty} a_n \cos\varsigma_n \xi + b_n \sin\varsigma_n \xi,$$

where $\varsigma_n = \frac{2n\pi}{\Delta}$,

$$a_0 = \frac{1}{\Delta}\int_0^{\Delta} f(\xi) d\xi, \qquad a_n = \frac{2}{\Delta}\int_0^{\Delta} f(\xi)\cos\varsigma_n \xi d\xi,$$

$$b_n = \frac{2}{\Delta}\int_0^{\Delta} f(\xi)\sin\varsigma_n \xi d\xi,$$

which is known as the Fourier series of f.

Proof: The proof of this theorem appears in most texts on Fourier series, (e.g. [Churchill1963] pp. 89 ff.). □

These conditions, for which $f(\xi)$ may be represented by its Fourier series, are known as the Dirichlet conditions. Note that there are many other sets of sufficient conditions.

Two alternative definitions of the series are

$$f(\xi) = A_0 + \sum_{n=1}^{\infty} A_n \cos(\varsigma_n \xi + \phi_n),$$

with $A_0 = a_0$,

$$A_n = \sqrt{a_n^2 + b_n^2},$$

$$\phi_n = \tan^{-1}(-b_n / a_n),$$

and the complex form

$$f(\xi) = \sum_{n=-\infty}^{\infty} z_n e^{i\varsigma_n \xi},$$

$$\text{with } z_n = \frac{1}{2\Delta} \int_{-\Delta}^{\Delta} f(\xi) e^{-i\varsigma_n \xi} d\xi$$

$$= A_n e^{i\phi_n}.$$

The two dimensional Fourier series of a space- and time-periodic function $f(x,t)$ with spatial period D and temporal period T is given by

$$f(x,t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} z_{mn} e^{i(k_m x + \omega_n t)},$$

$$\text{with } \omega_n = \frac{2n\pi}{T},\ k_m = \frac{2m\pi}{D},$$

$$\text{and } z_{mn} = \frac{1}{4DT} \int_{-D}^{D} \int_{-T}^{T} e^{-i(k_m x + \omega_n t)} f(x,t)\, dt\, dx.$$

Fourier series of many functions appear in tables, such as the CRC standard mathematical tables [CRC1978], and Gradshteyn and Ryzhik's tables of integrals, series and products [Gradshteyn1965]. When the function is not a standard form, it may be a shifted version of a standard one. In such a case the following theorem is helpful.

THEOREM A.2: *(Shift Theorem) If* $f(\xi)=a_0+\sum_{n=1}^{\infty} a_n\cos\varsigma_n\xi+b_n\sin\varsigma_n\xi$, *then*

$$f(\xi-\delta)=a_0'+\sum_{n=1}^{\infty} a_n'\cos\varsigma_n\xi+b_n'\sin\varsigma_n\xi \ ,$$

with $a_0'=a_0$,

$$a_n'=a_n\cos\varsigma_n\delta-b_n\sin\varsigma_n\delta \ ,$$

$$b_n'=b_n\cos\varsigma_n\delta+a_n\sin\varsigma_n\delta \ .$$

Proof: Using the second definition of the series, $f(\xi)=A_0+\sum_{n=1}^{\infty} A_n\cos(\varsigma_n\xi+\phi_n)$.

Clearly, $f(\xi-\delta)=A_0+\sum_{n=1}^{\infty} A_n\cos(\varsigma_n(\xi-\delta)+\phi_n)=A_0+\sum_{n=1}^{\infty} A_n\cos(\varsigma_n\xi+(\phi_n-\varsigma_n\delta))$.

Converting this back to the form of the original series,

$$a_0'=A_0=a_0 \ ,$$

$$a_n'=A_n\cos(\phi_n-\varsigma_n\delta)=A_n(\cos\phi_n\cos\varsigma_n\delta+\sin\phi_n\sin\varsigma_n\delta) \ ,$$

$$=A_n(\cos(\tan^{-1}(-b_n/a_n))\cos\varsigma_n\delta+\sin(\tan^{-1}(-b_n/a_n))\sin\varsigma_n\delta) \ ,$$

$$=\sqrt{a_n^2+b_n^2}\left(\frac{\cos\varsigma_n\delta}{\sqrt{1+b_n^2/a_n^2}}+\frac{(b_n/a_n)\sin\varsigma_n\delta}{\sqrt{1+b_n^2/a_n^2}}\right) \ ,$$

$$=a_n\cos\varsigma_n\delta-b_n\sin\varsigma_n\delta \ .$$

The expression for b_n' follows in a similar manner. □

DEFINITION A.1: *The* **Fourier Transform** *of a function* $f(\xi)$ *is*

$$\mathcal{F}[f(\xi),\varsigma]=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(\xi)e^{i\varsigma\xi}d\xi \ .$$

The **Fourier cosine transform** *is* $\mathcal{F}_c[f(\xi),\varsigma]=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(\xi)\cos\varsigma\xi d\xi$.

The **Fourier sine transform** *is* $\mathcal{F}_s[f(\xi),\varsigma]=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(\xi)\sin\varsigma\xi d\xi$.

Note: $\mathcal{F}[f(\xi),\varsigma]=\mathcal{F}_c[f(\xi),\varsigma]+i\mathcal{F}_s[f(\xi),\varsigma]$ follows directly from the Euler relations, $e^{\pm i\theta}=\cos\theta\pm i\sin\theta$. The Fourier transform of a function of time characterizes its frequency content. In the same way as a Fourier series gives the series of amplitudes

at discrete frequencies from which any periodic function may be reconstructed, the transform gives the amplitudes at frequencies distributed continuously, from which a possibly aperiodic function may be reconstructed.

DEFINITION A.2: *The set of amplitudes in the series or the continuous transform function is referred to as the* **spectrum** *of the original function.*

In two dimensions, the transform is defined as

$$\mathcal{F}[f(x,t),k,\omega] = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,t)e^{i(kx+\omega t)}dtdx \quad ,$$

with the sine and cosine transforms defined analogously.
In Theorems A.1 and A.2, the Fourier transform of a function with a phase-shifted argument appears as part of each term in the sums. The following theorem relates the Fourier transform of a function to the transform of the same function with the phase shifted.

THEOREM A.3: *(Shift theorem for Fourier Transforms)*

1. $\mathcal{F}[f(\xi-\delta),\varsigma] = e^{i\varsigma\delta}\mathcal{F}[f(\xi),\varsigma]$
2. $\mathcal{F}_c[f(\xi-\delta),\varsigma] = \cos\varsigma\delta\mathcal{F}_c[f(\xi),\varsigma] - \sin\varsigma\delta\mathcal{F}_s[f(\xi),\varsigma]$
3. $\mathcal{F}_s[f(\xi-\delta),\varsigma] = \cos\varsigma\delta\mathcal{F}_s[f(\xi),\varsigma] + \sin\varsigma\delta\mathcal{F}_c[f(\xi),\varsigma]$

Proof: 1: The proof is quite straightforward; see, for example [Gaskill1978].
2 and 3: Since $\mathcal{F}[f(\xi),\varsigma] = \mathcal{F}_c[f(\xi),\varsigma] + i\mathcal{F}_s[f(\xi),\varsigma]$,

$$e^{i\varsigma\delta}\mathcal{F}[f(\xi),\varsigma] = \cos\varsigma\delta\mathcal{F}_c[f(\xi),\varsigma] + i\sin\varsigma\delta\mathcal{F}_c[f(\xi),\varsigma] + i\cos\varsigma\delta\mathcal{F}_s[f(\xi),\varsigma] - \sin\varsigma\delta\mathcal{F}_s[f(\xi),\varsigma] \quad .$$

Matching terms, relations 2 and 3 follow immediately. □

The following definition and theorem are related to filtering. They show two equivalent ways of changing the spectrum of a function.

DEFINITION A.3: *The* **convolution integral** *of $g(\xi)$ with a filter $f(\xi)$ is defined as*

$$f \circ g(\xi) = \int_{-\infty}^{\infty} g(\delta)f(\xi-\delta)d\delta \quad .$$

The effect of convolution is to change the spectrum, in the following way:

THEOREM A.4: *(Convolution Theorem) If the Fourier transforms of $f(\xi)$ and $g(\xi)$ exist, then $\mathcal{F}[f \circ g(\xi),\varsigma] = \mathcal{F}[f(\xi),\varsigma]\,\mathcal{F}[g(\xi),\omega]$.*

Proof: The proof of this theorem appears in most introductory texts on Fourier transforms. □

The significance of this theorem is that if $f(\xi)$ is a filter function, designed to enhance or remove high frequency information from $g(\xi)$, then its ability to do so may be determined immediately from the transform of f. The amount of information passed by the filter at any frequency is proportional to the value of the transform at that frequency. Practical filtering in computer graphics involves filters which are non-zero over a small region, so that the integral may be computed cheaply.

For reference, the transforms of several functions which are used as examples appear below.

Name	$f(\xi)$	$\mathcal{F}[f(\xi),\varsigma]$
δ function	$\delta(\xi)$	$\frac{1}{\sqrt{2\pi}}$
Gaussian	$e^{-\xi^2/2\sigma^2}$	$\sigma e^{-\varsigma^2\sigma^2/2}$
Exponential	$e^{-\lambda\xi}$, $\lambda,\xi>0$	$\frac{1}{\sqrt{2\pi}}\frac{\lambda+i\varsigma}{\lambda^2+\varsigma^2}$

References

CRC1978 CRC, *CRC Standard Mathematical Tables*, CRC Press, Boca Raton (1978).

Adage1982 Adage Corporation, *IKONAS RDS 3000 Programmers Reference Manual*. 1982.

Numerical1988 Numerical Algorithms Group, *NAG Fortran Library Introductory Guide, Mark 13*. 1988.

Barlow1958 H.B. Barlow, Temporal and spatial summation in human vision at different background intensities, *Journal of Physiology* **141** pp. 337-350 (1958).

Barros1979 J. Barros and H. Fuchs, Generating smooth 2-D mono color line drawings on video displays, *Computer Graphics* **13**(2) pp. 260-269 (August 1979).

Blakemore1969 C. Blakemore and F.W. Campbell, On the existence of neurons in the visual system selectively sensitive to the orientation and size of retinal images, *Journal of Physiology* **203** pp. 237-260 (1969).

Blyth1986 T.S. Blyth and E.F. Robertson, *Matrices and Vector Spaces*, Chapman and Hall, London (1986).

Braddick1974 O.J. Braddick, Low-level and high-level processes in apparent motion, *Philosophical Transactions of the Royal Society of London B* **290** pp. 137-151 (1974).

Burr1979 D.C. Burr, On the visibility and appearance of objects in motion, D. Phil Thesis, Cambridge (1979).

Campbell1968 F.W. Campbell and J.G. Robson, Application of Fourier analysis to the visibility of gratings, *Journal of Physiology* **197** pp. 551-566 (1968).

Catmull1974 E. Catmull, *A Subdivision Algorithm for Computer Display of Curved Surfaces*, University of Utah PhD Thesis 1974.

Catmull1979 E. Catmull, A tutorial on compensation tables, *Computer Graphics* **13**(2) pp. 1-7 (August 1979).

Churchill1963 R.V. Churchill, *Fourier Series and Boundary Value Problems*, McGraw-Hill, London (1963).

Conn1989 A.R. Conn, *personal communication*. 1989.

Connor1982 J.D. Connor, The temporal properties of rod vision, *Journal of Physiology* **332** pp. 139-155 (November 1982).

Cook1984 R. Cook, T. Porter, and L. Carpenter, Distributed ray tracing, *Computer Graphics* **18**(3) pp. 137-145 (July 1984).

Cook1986 R.L. Cook, Stochastic sampling in computer graphics, *ACM Transactions on Graphics* **5**(1) pp. 51-72 (January 1986).

Crow1976 F.C. Crow, *The Aliasing Problem in Computer Synthesised Shaded Images*, Dept. of Computer Sc., University of Utah (March 1976). PhD. Thesis. Also available as University of Utah Technical Report UTEC-CSc-76-015

Crow1981 F.C. Crow, A comparison of anti-aliasing techniques, *IEEE Computer Graphics and Applications* **1**(1)(Jan. 1981).

Davis1979 P.J. Davis, *Circulant Matrices*, John Wiley & Sons, Inc, Toronto (1979).

Dierckx1977 P. Dierckx, An algorithm for least-squares fitting of cubic spline curves to functions on a rectilinear mesh over a rectangle, *Journal of Computational and Applied Mathematics* **3**(2) pp. 113-129 (1977).

Dippé1985 M.A. Dippé and E.H. Wold, Anti-aliasing through stochastic sampling, *Computer Graphics* **19**(3) pp. 69-78 (July 1985).

Dwyer1948 P.S. Dwyer and M.S. McPhail, Symbolic Matrix Derivatives, *Annals of Mathematical Statistics* **19** pp. 517-534 (1948).

Fahle1981 M. Fahle and T. Poggio, Visual hyperacuity: spatiotemporal interpolation in human vision, *Proceedings of the Royal Society of London B* **213** pp. 451-477 (1981).

Feibush1980 E. Feibush, M. Levoy, and R. Cook, Synthetic texturing using digital filters, *Computer Graphics* **14**(3) pp. 294-301 (July 1980).

Foley1983 J. Foley and A. van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading Mass. (1983).

Forsey1990 D.R. Forsey, *Modelling Articulated Figures for Computer Animation*, University of Waterloo Doctoral Dissertation, in progress 1990.

Gaskill1978 J.D. Gaskill, *Linear Systems, Fourier Transforms and Optics*, John Wiley, New York (1978).

Golub1983 G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD (1983).

Gradshteyn1965 I.S. Gradshteyn and I.M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, London (1965).

Graham1981 A. Graham, *Kronecker Products and Matrix Calculus with Applications*, Ellis Horwood, Chichester (1981).

Kajiya1981 J. Kajiya and M. Ullner, Filtering high quality text for display on raster scan devices, *Computer Graphics* **15**(3) pp. 7-15 (Aug. 1981).

Kaufman1974 L. Kaufman, *Sight and Mind*, Oxford University Press, New York (1974).

Kelly1979 D.H. Kelly, Motion and vision I. stabilized images of stationary gratings, *Journal of the Optical Society of America* **69** pp. 1266-1274 (1979).

Kelly1979a D.H. Kelly, Motion and vision II. stabilized spatiotemporal threshold surface, *Journal of the Optical Society of America* **69** pp. 1340-1349 (1979).

Kelly1982 D.H. Kelly, Motion and vision IV. isotropic and anisotropic spatial responses, *Journal of the Optical Society of America* **72** pp. 432-439 (1982).

Koonderink1979 J.J. Koonderink and A.J. van Doorin, *Optics Letters* **4**(1) pp. 32-34 (1979).

Lee1985 M. Lee, R. Redner, and S. Uselton, Statistically optimized sampling for distributed ray tracing, *Computer Graphics* **19**(3) pp. 61-68 (July 1985).

Mandelbrot1977 B.B. Mandelbrot, *Fractals, Form, Chance and Dimension*, W.H. Freeman and co., San Francisco (1977).

Mitchell1987 D.P. Mitchell, Generating antialiased images at low sampling densities, *Computer Graphics* **21**(4) pp. 65-72 (July 1987).

Mitchell1988 D.P. Mitchell and A.N. Netravali, Reconstruction filters in computer graphics., *Computer Graphics* **22**(4) pp. 221-228 (August 1988).

Nielson1988 K.R.K. Nielson and B.A. Wandell, *Journal of the Optical Society of America A* **5**(5) pp. 743-755 (1988).

Oppenheim1981 A.V. Oppenheim and J. Lim, The importance of phase in signals, *Proceedings of the IEEE* **69**(5) pp. 528-541 (May 1981).

Purgathofer1987 W. Purgathofer, A Statistical Model for Adaptive Stochastic Sampling, *Computers and Graphics* **11**(2) pp. 157-162 (1987).

Rau1965 C.R. Rau, *Linear Statistical Inference and Applications*, John Wiley and Sons, New York (1965).

Robson1966 J.C. Robson, Spatial and temporal contrast-sensitivity functions of the visual system, *Journal of the Optical Society of America* **56**(8) pp. 1141-1142 (1966).

Sachs1971 M. Sachs, J. Nachmias, and J. Robson, Spatial frequency channels in human vision, *Journal of the Optical Society of America* **61** pp. 1176-1186 (1971).

Smirnov1961 V.I. Smirnov, *Linear Algebra and Group Theory*, Dover Publications, New York (1961).

Stark1977 H. Stark, Diffraction patterns of nonoverlapping circular grains, *Journal of the Optical Society of America* **67**(5) pp. 700-703 (May 1977).

Thomas1975 J.P. Thomas, Spatial resolution and spatial interaction, pp. 233-263 in *Handbook of Perception*, ed. E.C. Carterette and M.P. Friedman, Academic Press, New York (1975).

Warnock1980 J.E. Warnock, The display of characters using gray level sample arrays, *Computer Graphics* **14**(3) pp. 302-307 (July 1980).

Watson1986 A.B. Watson, A. Ahumada, and J.E. Farrell, The window of visibility: A psychophysical theory of fidelity in time-sampled visual motion displays, *Journal of the Optical Society of America A* **3**(3) pp. 300-307 (March 1986).

Westheimer1975 G. Westheimer and S.P. McKee, Visual acuity in the presence of retinal image motion, *Journal of the Optical Society of America* **65**(7) pp. 847-850 (1975).

Whittaker1915 E.T. Whittaker, On the functions which are represented by the expansions of the interpolation theory, *Proceedings of the Royal Society of Edinburgh* **35** pp. 181-194 (1915).

Wilson1984 H.R. Wilson and D.J. Gelb, Modified line-element theory for spatial-frequency and width discrimination, *Journal of the Optical Society A* **1**(1) pp. 124-131 (January 1984).

Wittke1987 J.P. Wittke, Moiré considerations in shadow-mask picture tubes, *Proceedings of the Society for Information Display* **28**(4) pp. 415-418 (1987).

Yellott1982 J.I. Yellott, Spectral analysis of spatial sampling by photoreceptors: topological disorder prevents aliasing, *Vision Research* **22** pp. 1205-1210 (1982).