Velocity-tuned Spatio-temporal Interpolation and Approximation in Vision

by

Jonathan G. Bliss

B.S.E.E. Northeastern University (1979)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

Massachusetts Institute of Technology

December 6, 1985

©Jonathan G. Bliss, 1985

The author hereby grants to M.I.T. permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author

Department of Electrical Engineering and Computer Science

December 6, 1985

Certified by

Prof. Tomaso A. Poggio
Thesis Supervisor

Accepted by

Arthur C. Smith
Chairman, Departmental Graduate Committee

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

APR 11 1986

LIBRARIES - 1 - Archives
Velocity-tuned Spatio-temporal Interpolation and Approximation in Vision

by

Jonathan G. Bliss

Submitted to the Department of Electrical Engineering and Computer Science on December 6, 1985 in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering.

Abstract

This thesis fully develops the theoretical details of two- and three-dimensional velocity-tuned filters. The concept of effective velocity bandwidth is introduced: a bandlimited function which is translating at a constant velocity within the effective velocity bandwidth of a velocity-tuned filter can be passed by the filter, with little or no degradation.

The application of velocity-tuned filters to the reconstruction of temporally sampled images is also developed in detail. This application is motivated by the constant-velocity assumption, which asserts that over spatial and temporal scales of interest, motion in images represents rigid-body translation at constant velocity. It is shown that velocity-tuned filters perform reconstruction by smoothing the sampled image along lines of constant velocity in the space-time domain; rather than a simple temporal operator, the velocity-tuned filter is a spatio-temporal interpolation operator.

Taking an alternative approach, the task of reconstructing a sampled image is cast as an ill-posed problem. A regularized solution is derived, using a regularizing functional which minimizes the norm of the nth directional derivative of the image along constant-velocity lines. The regularized solution is shown to be a velocity-tuned approximation filter. In the space-time domain, the effect of the filter is convolve the image with an approximating spline along constant-velocity lines.

Two and three-dimensional velocity-tuned filters were implemented as computer-based image processing programs. It is shown that the velocity-tuned spatio-temporal approximation filter can be implemented as a very simple, local, and intuitively pleasing operator. The effects of the filters are demonstrated, first using continuous time-varying input images, and then in the context of reconstruction of sampled images. We illustrate the effect of disparity between velocities in the input image and the velocity to which the filter is tuned, as well as the effect of changing the effective velocity bandwidth of the filter.

Next, we consider the performance of the velocity-tuned filter in the presence of noisy inputs. In most cases, a velocity-tuned filter decreases the noise power of the input image, thus increasing the detectability of moving objects, as long as the velocity of the object is within the effective velocity bandwidth of the filter. A psychophysical experiment which demonstrates this effect in the human observer was performed.

Finally, the consequences of relaxing the constant-velocity assumption are explored, considering non-constant velocity motion, as well as temporal and spatial variation in velocity. In addition, we consider possible biological implementation schemes. We conclude with two possible applications, namely the reduction of transmission bandwidth of televideo signals, and the processing of noisy image sequences, such as the dynamic radiotracer studies done in Nuclear Medicine.

Thesis Supervisor: Tomaso A. Poggio
Professor of Psychology
MIT Artificial Intelligence Lab
Acknowledgments

I offer many, many thanks to Professor Poggio, whose guidance, patience, and friendship made this thesis possible. I'm not sure how, but he single-handedly managed to make regularization theory and psychophysics my allies, rather than adversaries. I will be forever indebted to him for showing me the difference between research and hacking computers.

I also thank Dr. Demetri Terzopoulos, for helping me overcome my mortal fear of variational calculus, for many stimulating discussions about regularization, and for proofreading those parts of this thesis which pertain to that subject.

Thanks also to Dr. Phillipe Brou who proofread the hairiest of the derivations which appear in the appendices, and provided good advice concerning the user interface in the psychophysical experiments.

I am also deeply indebted to my those friends who "volunteered" themselves to be subjects in my psychophysical experiments. I believe all but one of them have been released from the hospital now, although one accidentally saw a "C" moving across the Times Square display, and had to be put on tri-cyclic anti-depressants. To Philippe, Demetri, Steve Gander, Eric Tiffany, Dave Clemens, and Krystine Jankowski: I owe you one.

Finally, thanks to the Psi Foundation, and to the Becton & Dickinson Corporation, which have both funded parts of this research.

I would also like to thank some others who have contributed in less material ways to this thesis. In particular, thanks to faculty members of the MIT Department of Electrical Engineering and Computer Science, who provided timely guidance and motivation, and coached this thesis from my reluctant fingers: Professors Arthur C. Smith, Larry Frishkopf, and William Peake.

In addition, thanks to faculty and students of the Harvard-MIT Division of Health Science and Technology, Program in Medical Engineering and Medical Physics, with whom I have shared these past few years, and expect to share a few more. In particular, I thank Professor Jim Weaver for exceptionally sound advice and patience.

An undertaking such as this could not be completed without a support network. Mine was centered around a loving God, who was manifested as both Revealer and Sustainer throughout this work, and especially in the past few months. My family has also been an important source of support in this endeavor, even though at times I have detected a desire on their part that I start doing something useful, like looking for a job. The support of friends at the MIT AI Lab was also crucial; among other things, they made sure I ate at least one good meal a day. Finally, my friend Kristine has been an endless source of love, hope, and perhaps most importantly, perspective.

Dedication

This thesis is dedicated to my parents, Philip '37 and Ruth Bliss, and to my late grandfather, Gardner R. Alden '13, whose excitement about medical engineering was partially responsible for my choice to follow this path.
# Table of Contents

## Chapter 1  Introduction

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Interpolation</td>
<td>10</td>
</tr>
<tr>
<td>1.2</td>
<td>Sampling Theorems</td>
<td>11</td>
</tr>
<tr>
<td>1.3</td>
<td>Reconstruction</td>
<td>13</td>
</tr>
<tr>
<td>1.4</td>
<td>Sampling and Reconstruction in the Frequency Domain</td>
<td>14</td>
</tr>
<tr>
<td>1.5</td>
<td>Sampling and Interpolation in Visual Systems</td>
<td>17</td>
</tr>
<tr>
<td>1.6</td>
<td>Temporal Sampling in Visual Systems</td>
<td>20</td>
</tr>
<tr>
<td>1.7</td>
<td>Temporal Interpolation in the Visual System</td>
<td>21</td>
</tr>
<tr>
<td>1.8</td>
<td>Evidence for Spatio-temporal Interpolation: Apparent Motion</td>
<td>22</td>
</tr>
<tr>
<td>1.9</td>
<td>Acuity and Motion</td>
<td>23</td>
</tr>
<tr>
<td>1.10</td>
<td>Towards a Computational Theory: the Constant Velocity Assumption</td>
<td>24</td>
</tr>
</tbody>
</table>

## Chapter 2  Theoretical Aspects of Velocity-Tuned Filters

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Constant-Velocity Theorems</td>
<td>27</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Implications of the Constant-Velocity Theorems</td>
<td>33</td>
</tr>
<tr>
<td>2.2</td>
<td>Velocity-Tuned Filters</td>
<td>37</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Two-dimensional Filters</td>
<td>39</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Space-time Domain Description of Two-dimensional Filter</td>
<td>40</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Effects of the Two-dimensional Filter</td>
<td>43</td>
</tr>
<tr>
<td>2.2.4</td>
<td>Effective Velocity Bandwidth of the Two-dimensional Filter</td>
<td>51</td>
</tr>
<tr>
<td>2.2.5</td>
<td>Resolution Tradeoffs in Two-dimensional Case</td>
<td>54</td>
</tr>
<tr>
<td>2.2.6</td>
<td>Three dimensional Velocity-tuned Filter</td>
<td>56</td>
</tr>
<tr>
<td>2.2.7</td>
<td>Space-time Domain Description of Three-dimensional Filter</td>
<td>59</td>
</tr>
<tr>
<td>2.2.8</td>
<td>Effects of the Three-dimensional Filter</td>
<td>60</td>
</tr>
<tr>
<td>2.2.9</td>
<td>Effective Velocity Bandwidth of the Three-dimensional Filter</td>
<td>64</td>
</tr>
<tr>
<td>2.2.10</td>
<td>Resolution Tradeoffs in Three-dimensional Case</td>
<td>69</td>
</tr>
</tbody>
</table>
## Table of Contents

### Section 2.3 Reconstruction Using Velocity-Tuned Filters
- 2.3.1 Two-dimensional Case ........................................... 69
- 2.3.2 Reconstruction of Two-dimensional Constant-velocity Inputs ...... 70
- 2.3.3 Minimum Sampling Rate for Two-dimensional Case .................. 71
- 2.3.4 Reconstruction of Arbitrary Two-dimensional Functions ............. 75
- 2.3.5 When Two-dimensional Reconstruction Fails ........................ 76
- 2.3.6 Reconstruction With Three-dimensional Velocity-tuned Filters ...... 78
- 2.3.7 Reconstruction of Three-dimensional Constant-velocity Inputs ...... 79
- 2.3.8 Minimum Sampling Rate for Three-dimensional Case ................. 80
- 2.3.9 Reconstruction of Arbitrary Three-dimensional Functions ........... 84
- 2.3.10 When Three-dimensional Reconstruction Fails ....................... 85

### Section 2.4 A Regularized Solution to Spatio-temporal Approximation
- 2.4.1 Regularization of Ill-posed Problems ................................ 88
- 2.4.2 Regularization of the Edge-detection Problem ....................... 89
- 2.4.3 Regularization of the Spatio-temporal Approximation Problem ...... 91
- 2.4.4 Regularization of Three-dimensional Spatio-temporal Approximation 94
- 2.4.5 General Regularized Solution to Spatio-temporal Approximation .... 96
- 2.4.6 Relation Between $\lambda$ and Velocity ................................ 98
- 2.4.7 Relation to Butterworth Filters ................................... 103

### Section 2.5 Relaxing the Constant-Velocity Assumption
- 2.5.1 Rigid-Body Translation with Arbitrary Velocity ..................... 104
- 2.5.2 Constant-velocity Translation with Sinusoidal Modulation .......... 105
- 2.5.3 Constant Velocity Translation With Two Modulating Sinusoids ...... 106
- 2.5.4 Constant-velocity Translation with Arbitrary Modulation .......... 107
- 2.5.5 Three-dimensional Case .......................................... 108
- 2.5.6 Temporal Windowing .............................................. 109
- 2.5.7 Three-Dimensional Case .......................................... 110
- 2.5.8 Predictions for Real-World Motion ................................ 111

### Chapter 3 Examples of Velocity-tuned Filtering and Velocity-tuned Reconstruction

#### Section 3.1 Implementation Issues
- 3.1.1 Computational Effort ........................................... 129
- 3.1.2 Constant-velocity Theorem for Discrete Case ....................... 130

#### Section 3.2 Spatio-temporal Approximation Algorithm

#### Section 3.3 Examples of Two-dimensional Spatio-temporal Approximation
- 3.3.1 Constant-velocity Input ........................................ 133
- 3.3.2 Sampled Constant-velocity Inputs ............................... 134
- 3.3.3 Non-constant-velocity Input .................................... 135

#### Section 3.4 Examples of Three-dimensional Spatio-temporal Approximation
- 3.4.1 Constant-velocity Inputs ........................................ 140
- 3.4.2 Velocity-Tuned Reconstruction .................................. 141

#### Section 3.5 Objects Moving in the Presence of Noise

129

135

140

141

150

151

152

152

153

156
Chapter 4  A Psychophysical Experiment: Detection of Moving Objects in the Presence of Noise  

<table>
<thead>
<tr>
<th>Section 4.1</th>
<th>Spatio-temporal Interpolation and Noise</th>
<th>159</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 4.2</td>
<td>Methods</td>
<td>161</td>
</tr>
<tr>
<td>Section 4.3</td>
<td>Results and Discussion</td>
<td>167</td>
</tr>
</tbody>
</table>

Chapter 5  Discussion  

<table>
<thead>
<tr>
<th>Section 5.1</th>
<th>Reconstruction in Real Time-varying Images</th>
<th>176</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.1</td>
<td>Velocity-Tuned Channels</td>
<td>176</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Spatially Local Spatio-temporal Approximation</td>
<td>177</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Velocity Measurement</td>
<td>178</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Motion Discontinuity</td>
<td>179</td>
</tr>
<tr>
<td>5.1.5</td>
<td>Causality</td>
<td>180</td>
</tr>
<tr>
<td>Section 5.2</td>
<td>Velocity Measurements</td>
<td>182</td>
</tr>
<tr>
<td>Section 5.3</td>
<td>Motion Smear</td>
<td>186</td>
</tr>
<tr>
<td>Section 5.4</td>
<td>Applications</td>
<td>187</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Edge-detection in Noisy Image Sequences</td>
<td>187</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Reduction of Transmission Bandwidth</td>
<td>189</td>
</tr>
<tr>
<td>Section 5.5</td>
<td>Conclusion</td>
<td>190</td>
</tr>
</tbody>
</table>

Appendix A  Impulses  

<table>
<thead>
<tr>
<th>Section A.1</th>
<th>One-dimensional Case</th>
<th>191</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section A.2</td>
<td>Two-dimensional Case</td>
<td>193</td>
</tr>
<tr>
<td>Section A.3</td>
<td>Discrete Case</td>
<td>196</td>
</tr>
</tbody>
</table>

Appendix B  Sampling Theorems  

<table>
<thead>
<tr>
<th>Section B.1</th>
<th>One-dimensional Functions</th>
<th>197</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section B.2</td>
<td>Aliasing</td>
<td>203</td>
</tr>
<tr>
<td>Section B.3</td>
<td>Implicit Reconstruction</td>
<td>206</td>
</tr>
<tr>
<td>Section B.4</td>
<td>Two-dimensional Sampling and Reconstruction</td>
<td>206</td>
</tr>
</tbody>
</table>

Appendix C  Constant-velocity Theorems  

<table>
<thead>
<tr>
<th>Theorem C.1</th>
<th>Two-dimensional Continuous Case</th>
<th>212</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theorem C.2</td>
<td>Three-dimensional Continuous Case</td>
<td>214</td>
</tr>
<tr>
<td>Theorem C.3</td>
<td>Duality for Two-dimensional Continuous Case</td>
<td>216</td>
</tr>
<tr>
<td>Theorem C.4</td>
<td>Duality for Three-dimensional Continuous Case</td>
<td>217</td>
</tr>
<tr>
<td>Theorem C.5</td>
<td>Duality for Alternative Three-dimensional Continuous Case</td>
<td>218</td>
</tr>
<tr>
<td>Theorem C.6</td>
<td>Two-dimensional Discrete Case</td>
<td>220</td>
</tr>
</tbody>
</table>
Appendix D. Velocity Tuned Filters ........................................... 225

Theorem D.1 Two-dimensional case ........................................ 225
Theorem D.2 Three-dimensional Case ..................................... 227
Theorem D.3 Alternate Three-dimensional Case ...................... 229

Appendix E. Reconstruction with Velocity-Tuned Filters .......... 231

Theorem E.1 Bandwidth Constraint in Two-Dimensional Case ...... 231
Theorem E.2 Minimum Sampling Rate in Two-dimensional Case ... 233
Theorem E.3 Arbitrary Bandlimited Two-dimensional Functions ... 235
Theorem E.4 Bandwidth Constraint in Three-dimensional Case .... 237
Theorem E.5 Minimum Sampling Rate for Three-dimensional Case.. 240
Theorem E.6 Arbitrary Bandlimited Three-dimensional Functions . 242

Appendix F. Regularized Filters ............................................. 245

Theorem F.1 Two-Dimensional Regularized Filter ................... 245
Theorem F.2 Two-Dimensional Case With Higher-Order Stabilizer 248
Theorem F.3 Three-Dimensional Case .................................... 251
Theorem F.4 Fourier Transform of Regularized Filter ............... 254

Appendix G. Arbitrary Motion in the Frequency Domain .......... 266

Theorem G.1 General Case .................................................. 266
Theorem G.2 Sinusoidal Modulation ...................................... 267
Theorem G.3 Sinusoidal Modulation in Two Dimensions ............ 269
Theorem G.4 Temporal Windowing – Two-dimensional Case ........ 271
Theorem G.5 Temporal Windowing – Three-dimensional Case ....... 272

References ............................................................................. 274
List of Figures and Plates

Chapter 2  Theoretical Aspects of Velocity-Tuned Filters

Section 2.1  Constant-Velocity Theorems
Figure 2.1.1  Constructing a two-dimensional constant-velocity function
Figure 2.1.2  Constant-velocity function in the frequency domain
Figure 2.1.3  Three-dimensional constant-velocity function
Figure 2.1.4  Sampling and reconstruction of constant-velocity functions
Figure 2.1.5  Low-pass filter reconstruction fails at lower sampling rates
Figure 2.1.6  Reconstruction of constant-velocity function with velocity-tuned filter

Section 2.2  Velocity-Tuned Filters
Figure 2.2.1  A rectangular pulse velocity-tuned filter
Figure 2.2.2  Rectangular velocity-tuned filter in space-time
Figure 2.2.3  The effect of velocity-tuned filter operation
Figure 2.2.4  Effect of velocity-tuned filter on constant-velocity inputs
Figure 2.2.5  Effect of velocity-tuned filter on arbitrary inputs
Figure 2.2.6  Critical point for determining velocity bandwidth
Figure 2.2.7  Critical point for passing arbitrary function
Figure 2.2.8  Support of three-dimensional velocity-tuned filter
Figure 2.2.9  Operation of three-dimensional velocity-tuned filter
Figure 2.2.10  Determining velocity bandwidth of three-dimensional filter
Figure 2.2.11  Three-dimensional bandwidth requirements for arbitrary input

Section 2.3  Reconstruction Using Velocity-Tuned Filters
Figure 2.3.1  Velocity-tuned filter "connects the dots" between sample values
Figure 2.3.2  Reconstruction of constant-velocity function with velocity-tuned filter
Figure 2.3.3  Recon. of arbitrary bandlimited function with velocity-tuned filter
Figure 2.3.4  3D velocity-tuned filter "connects the dots" between sample values
Figure 2.3.5  Reconstruction of 3D constant-velocity function with velocity-tuned filter
Figure 2.3.6  Recon. of arbitrary 3D bandlimited function with velocity-tuned filter

Section 2.4  A Regularized Solution to Spatio-temporal Approximation
Figure 2.4.1  Regularized Filters of Orders 1.2 and 3
Figure 2.4.2  Frequency Spectra of Filters of Orders 1.2 and 3
Figure 2.4.3  Second Degree Regularized Filter with Increasing Lambda
Figure 2.4.4  Frequency Spectrum of Second Degree Filter with Increasing Lambda

Section 2.5  Relaxing the Constant-Velocity Assumption
Figure 2.5.1  Constant-velocity translation with sinusoidal modulation
Figure 2.5.2  Sinusoidal modulation of bandlimited function
Figure 2.5.3  Two-tone modulation – Main sidebands for $A_2 >> A_1$
Chapter 3  Examples of Velocity-tuned Filtering and Velocity-tuned Reconstruction

Section 3.1  Implementation Issues

Figure 3.1.1  Constant-velocity Mod-M Rotation of Discrete Function
Figure 3.1.2  Discrete Fourier Transform of Function in Figure 3.1.1

Section 3.3  Examples of Two-dimensional Spatio-temporal Approximation

Figure 3.3.1  Legend for Plates 1 and 2
Figure 3.3.2  Plate 1: Constant-velocity Inputs
Figure 3.3.3  Plate 2: Non-constant Velocity Inputs

Section 3.4  Examples of Three-dimensional Spatio-temporal Approximation

Figure 3.4.1  Plate 3: Constant-velocity Inputs
Figure 3.4.2  Plate 4: Reconstruction of Constant-velocity Inputs
Figure 3.4.3  Plate 5: Effect of Velocity-tuning on Reconstruction
Figure 3.4.4  Plate 6: Two-dimensional Velocity Disparities

Section 3.5  Objects Moving in the Presence of Noise

Figure 3.5.1  Plate 7: Velocity-tuned Filter With Noisy Input Sequence

Chapter 4  A Psychophysical Experiment: Detection of Moving Objects in the Presence of Noise

Figure 4.3.1-5  Experimental Results for Human Observers

Chapter 5  Discussion

Section 5.2  Velocity Measurements

Figure 5.2.1  Approximate Support of Y-cell Operator
Figure 5.2.1  Y-cell Operator is Implicit Velocity-Tuned Filter

Appendix B.  Sampling Theorems

Figure B.1.1  Sampling as Viewed in the Frequency Domain
Figure B.1.2  Reconstruction of Original Function From Its Samples
Figure B.2.1  Aliasing and Partial Recovery of Undersampled Functions
Figure B.2.1  Rectangular Sampling Grid
Figure B.2.2  Hexagonal Sampling Grid

Appendix E.  Reconstruction with Velocity-Tuned Filters

Figure E.1.1  Reconstruction of constant-velocity function with velocity-tuned filter
Figure E.3.1  Recon. of arbitrary bandlimited function with velocity-tuned filter
Figure E.4.1  Reconstruction of 3D constant-velocity function with velocity-tuned filter
Figure E.6.1  Recon. of arbitrary 3D bandlimited function with velocity-tuned filter
Chapter 1

Introduction

The last time you went to the movies to relax, your visual system did a rather remarkable thing. Even though the projectionist presented to you a sequence of discrete, static images, your perception was that of watching a continuous, time-varying image. In addition, although you probably weren't aware of it at the time, this transformation took place with little or no loss of acuity: the edges of things moving in the image probably looked sharp and distinct, rather than fuzzy or smeared. Of course, there are processes going on at many levels of the visual system when you watch a movie, ranging from low-level processes, such as locating edges in the image, through higher-level processes, such as recognizing the face of your favorite actor. This panoply of interacting processes makes it dangerous to draw conclusions about any single process on the basis of the general movie-viewing experience. However, the transformation of discrete images into continuous motion is a clue that something interesting is happening, and compels one to examine the properties of this transformation, and that is the intent of this thesis. The examination will begin with a few known and easily demonstrated psychophysical results that more clearly define the nature of the transformation. We will then continue, with a theoretical analysis of what the task is, and how it can be performed with the information available to the visual system. Finally, we will attempt to verify a prediction made by the theory, with a new psychophysical experiment.
1.1 Interpolation

What the human visual system is doing while you watch a movie is interpolating the moving image. That is, given discrete samples of the continuous image, the visual system in some sense computes what happens in the image between the samples. The mathematics of the interpolation task, especially in conjunction with the sampling process, has been well studied, and we now briefly discuss some of the salient results.

We begin with a more precise definition. By interpolation we mean a process by which, given values of a function at some set of points, we generate additional values of the function between those “known points”. In general, the known values could be either a set of discrete points, or a set of continuous segments, although we will be dealing exclusively with the former case. The result of interpolation can be either a discrete or continuous function. For example, a point in a discrete result could be formed by computing a weighted average of the known values in the vicinity of that point. A continuous interpolation result is typically found by computing a weighted sum of continuous functions, referred to as interpolating functions; with the result in hand, one can then compute the value of the function at all points between the known values.

It should be noted that, as defined, the interpolation task does not have a unique solution. For example, the set of continuous functions that can be drawn through a set of \(N\) points contains an infinite number of members; any polynomial with \(N\) or more freely variable coefficients would suffice. Thus, we must add constraints to the interpolation problem to select “the best” of the possible interpolation solutions. For example, we might select from the set of possible interpolated results, the one which is the smoothest function, under some criterion of smoothness.

1.2 Sampling Theorems

Probably the most common context in which one finds interpolation is in the problem of sampling and reconstruction of a signal. The problem is stated as follows: Given some continuous function \(f(\alpha)\), we must generate a set of discrete samples \(f_n[n]\), which contains all of the “information” available
in $f(\alpha)$. Without delving into the realm of information theory any more than necessary, let us merely state that if the samples capture all of the information in a function, this implies that the original function can be exactly recovered, or reconstructed from those samples. Reconstruction of the original function clearly requires interpolation of the sampled function; now we have added the constraint that the interpolated result should exactly equal the original function.

This problem can be broken down into two questions: (1) What class of functions, if any, can be completely characterized by a set of discrete samples; and (2) how can the original signal be reconstructed from those samples. These questions are addressed by a set of theorems known collectively as sampling theorems. Probably the most commonly encountered is the classic sampling theorem, which had its origins with Whittaker [1], and which was stated by Shannon [2] as follows. "If a function $f(t)$ contains no frequencies higher than $W$ cycles per second, it is completely determined by giving its ordinates at a series of points spaced $1/2W$ apart." As a matter of terminology, when all frequency components of $f(t)$ have frequency less than $W$, we say that $f(t)$ is bandlimited to $W$.

The sampling theorem implies that if we have a function $f(t)$ which is bandlimited to $W$, and sample it at evenly-spaced points $t = nT$, and when the sampling rate $1/T$ equals twice the frequency of the highest frequency component, then all the information in the function has been captured, and the original function can be recovered from these samples.

As noted by Shannon, the Sampling Theorem really states that a bandlimited function $f(\alpha)$, over some interval $\alpha_0 \leq \alpha < \alpha_0 + J$, has only $2JW$ degrees of freedom, and can thus be completely specified by $2JW$ numbers. One such set of numbers is obtained by regular sampling, i.e. sampling at evenly spaced points, at a rate of $2W$ samples per unit $\alpha$, over the interval $J$, but this is not the only set. One can, for instance, sample at an irregular rate, as long as the average sampling rate over $J$ is $2W$. Alternatively, one can sample at half this rate, generating two numbers per sample point, say the value of the function, and its derivative. As long as $2JW$ independent numbers are produced, the information in the original function will be theoretically captured. Implementation of these alternative approaches tend to have problems however. For example, it is usually difficult to calculate derivatives
with the precision required to get good reconstruction [2]. Consequently, alternative schemes tend to be less practically useful than those based on regular sampling.

Clearly, if $2JW$ samples contain the information in a bandlimited function, then any greater number of samples will do so as well. Thus, we can relax the sampling theorem by saying that regular sampling of a bandlimited function, at a rate greater than or equal to $2W$, a quantity called the Nyquist rate, will capture the information in the function. From Shannon's argument, we see that any number greater than $2JW$ samples is redundant, i.e., they don't add any new information about the function. However, in cases where there is some uncertainty about the sample values, for instance in the presence of noise, one can only estimate what the actual sample values are. In this case, the redundancy introduced by oversampling provides information which can be used to reduce the uncertainty in these estimates.

1.3 Reconstruction

The sampling theorem answers the first of our two questions. Now we must deal with the second; having the set of sample values in hand, how do we reconstruct the original function?

Shannon also showed [2] how to reconstruct the original function from its samples, noting that the same result had been given by mathematicians, notably Whittaker [1], some fifteen years previously. The result is that $f(t)$ can be reconstructed exactly by computing a weighted average of time-shifted interpolating functions which have the form of $\sin x/x$, or sinc functions; the weighting factors are the sample values themselves. Thus, letting $f(nT)$ denote the sample values of $f(t)$ at times $t = nT$:

$$f(t) = \sum_{n=0}^{\infty} f(nT) \frac{\sin(\pi/T)(t - nT)}{(\pi/T)(t - nT)}$$

1.3.1

This result may not be intuitive, but it is correct. It implies that and bandlimited function can be expressed as the weighted sum of sinc functions; i.e., sinc functions form a basis for the space of
Section 1.4

Sampling and Reconstruction in the Frequency Domain

bandlimited functions. We note that the sinc functions above are all zero at the sample locations \( t = kT \), except when \( k = n \), in which case we get \( \text{sinc}(0) \) which has unity value. Thus, at the values of \( t \) where the sample points occur, \( f(t) = f(nT) \), which implies that the value of the reconstructed function equals the value of the sampled function. Thus if the sampled function and reconstructed function were drawn on the same ordinate, the reconstructed function goes through all the sample points, which of course it should. This demonstrates that the reconstruction operation described here is an interpolation operation; the result retains the sample values, and fills in the values in between them.

1.4 Sampling and Reconstruction in the Frequency Domain

Perhaps the most elegant proofs of sampling and reconstruction theorems occur in the frequency domain; one such proof is outlined in Appendix B. There are a few important points which should be summarized here because they lay the groundwork for the theoretical development in the next chapter.

Of course, using the frequency domain requires that \( f(t) \) has a frequency-domain representation. For bandlimited functions, we refer of course to \( F(\omega) \), the Fourier transform or frequency spectrum of \( f(t) \). Alternatively, if \( f(t) \) is to represent some real-world signal, such as light intensity falling on a photoreceptor, it must be considered as a stochastic or random process, for which the Fourier transform does not exist. In this case, one can use the power spectral density of \( f(t) \) to derive a similar sampling theorem, which states that right-hand side of Equation 1.3.1 converges to \( f(t) \) in a probabilistic sense [3]. For the theoretical development of the next chapter, we will assume, unless otherwise stated, that \( f(t) \) is a function whose Fourier transform exists, remembering however, that this is an ideal-case assumption, which can only tell us the optimum performance level to be expected in sampling and reconstruction tasks.

The major concepts from the frequency domain are as follows. Sampling \( f(t) \) results in replication of the Fourier transform, or spectrum of \( f(t) \) along the temporal frequency axis. That is, exact copies of \( F(\omega) \) are found evenly-spaced along the \( \omega \) axis of the Fourier transform of the sampled function. The spacing between replicates equals the sampling frequency, \( 2\pi/T \). Now, if \( f(t) \) is bandlimited
to $W$, then $F(\omega) = 0$ for $|\omega| > W$. Thus, if the sampling rate is high enough, then the spectral replicates will be spaced sufficiently far apart that they will not overlap each other. In principle then, the original spectrum can be isolated and recovered from its replicates; then the original function will also be recovered, due to the uniqueness property of Fourier transforms. If the sampling rate is too low, or if the original function is not bandlimited, then the spectral replicates overlap, and because they get added together, it becomes impossible to recover the original spectrum.

Now, we are ready to see the ties to the sampling theorem. A function which is bandlimited to $W$ has a spectrum which is non-zero for $-W < \omega < W$, thus covering an interval that is $2W$ long, along the frequency axis. Thus, to preclude overlap of the spectral replicates, we need the spacing between replicates to be greater than or equal to $2W$. Since the spacing between the replicates equals the sampling frequency, $f_s = 1/T$, we get Shannon's first result immediately; the original spectrum can be recovered exactly if the sampling frequency is greater than or equal to twice the highest frequency component of the original function, i.e. the Nyquist rate. It is implied that the sample values thus obtained contain all the information in the original function, otherwise exact reconstruction would not be possible.

Shannon's second result is a little trickier to produce; those who are unfamiliar with the concepts below are directed once again to Appendix B. Assuming we have sampled a bandlimited function at or above the Nyquist rate, the original spectrum can be recovered from its spectral replicates by passing the sampled function through a linear filter, which has unity gain for those frequencies included in the original function, and zero gain elsewhere. Such a filter would then effectively multiply the original spectrum by one, passing it unchanged, and the spectral replicates by zero, totally eliminating them. Thus, we would say that the frequency response, $H(\omega)$, of the filter must have unity value for $|\omega| < W$, and zero outside this range, in order to recover the original spectrum. This type of filter is called an ideal low-pass filter, since it passes only frequencies between zero and some cutoff-frequency. As the name implies, it is not physically realizable, although it can be well approximated. Now, the effect of this filtering operation in the time-domain is the convolution of the sampled function with the inverse Fourier transform of the frequency response, which happens to be a sinc function. Moreover, when
one evaluates the convolution integral, one obtains the same expression as above, expressing $f(t)$ as the weighted sum of time-shifted sinc functions.

At this point, the mathematics are not the key issue. What is important is that the original function is recovered from its samples by using a low-pass filtering operation, which has the effect of smoothing or smearing any function to which it is applied.

An important observation is that in some systems, interpolation occurs without an explicit interpolation step. If a sampled function is passed through a filter whose system function has arbitrary shape, but which is sufficiently bandlimited to eliminate the spectral replicates from Fourier transform of the sampled function, then reconstruction occurs implicitly. Of course, the output of the filter will not be the same as the original, unsampled function unless the filter has constant gain over the support of the original spectrum; rather, the effect of the filter is to reconstruct the original function, and then "filter it some more", by multiplying its spectrum by the system function of the filter. Thus, the output of the filter will be the same whether the input is the original function, or samples of the original function; this fact illustrates the concept that the samples contain all the information of the original function. We note that this is true even if the original function was sampled below the Nyquist rate; the elimination of the spectral replicates is all that is required, although if the sampling rate is too low, even this will be impossible. As an example, consider the Gaussian filter; this filter is often used for smoothing images as part of the edge-detection task in vision systems [4,5]. Both the impulse response and system function of this filter have the shape of the Gaussian function; although this function never reaches zero, it approaches zero very rapidly, and thus approximates a bandlimited filter, whose bandwidth is controlled by the parameter $\sigma$. Therefore, the result of passing an arbitrary, continuous bandlimited function through a Gaussian filter can be nearly indistinguishable from the result of passing the sample values of that function through the same filter.

It should be noted finally that there are a large compendium of sampling errors, such as aliasing and jitter, that can result in degradation of the reconstructed function. These issues are dealt with at length in the literature [6-8], but although they are certainly relevant in vision systems, they will not
be emphasized in this thesis. In general, we will be presenting ideal “best-case” scenarios, with the understanding that these represent what can be achieved ideally, but not practically.

1.5 Sampling and Interpolation in Visual Systems

The very first thing the human visual system, or in fact any computer-vision system does is sample the continuous light intensity distribution which falls on the retina, using a two-dimensional array of photoreceptors. Therefore it is clear that issues related to sampling are extremely relevant to vision. Thus, one can ask how to what degree the criteria developed in the sampling theorem are met by a particular vision system.

We note that one-dimensional results presented above must be extended to discuss this two-dimensional situation; we will not derive them explicitly, but merely state them here. In the case where the (two-dimensional) spectrum of the original function is circularly bandlimited to \( W \), i.e. is zero outside a circular region of the two-dimensional frequency plane, whose radius is \( W \), and with the hexagonal arrangement of photoreceptors such as is found in the retina, a familiar result appears. If the spacing \( D \) between the centers of neighboring receptors is small enough that the effective sampling rate \( 1/D \) is greater than or equal to \( 2W \), then the sampled function can be recovered exactly from its samples. More details on hexagonal sampling can be found in [9,10].

In the case of the human visual system, the intensity function falling on the retina is circularly bandlimited by the optics of the eye to about 60 cycles per degree [11], and the sampling rate in the fovea is around 120 to 130 samples per degree, thus satisfying these criteria. Outside the fovea, the sampling rate falls below the Nyquist rate, so these criteria are not met.

However, it seems clear that the original intensity function is not reconstructed by the visual system; if it were, there would be no way to store or manipulate it. Indeed, it would be uncharacteristically inefficient to perform a reconstruction; if all of the information in the intensity function is captured
in a finite set of samples, then it is clearly easier and thus "smarter" to process the samples than to process the whole function.

Nevertheless, there is irrefutable evidence that the human visual system does interpolate at least some of the retinal samples to obtain a finer-grain representation of the retinal image; whether or not this interpolation is explicit or implicit is less clear. In particular, we refer to experiments which demonstrate hyperacuity in humans. The classic test for measuring visual acuity is to present two points of light, side by side, and decrease the distance between them until they appear to be a single point. Hyperacuity is the ability to detect smaller offsets than can be detected in the classic two-point discrimination test, which arises in some situations; for example, see Westheimer [12] and Westheimer and McKee [13]. The limit of two-point acuity is about 1' of arc, as measured by Westheimer, and decreases dramatically to 2" to 5" of arc (an order of magnitude smaller) in some hyperacuity experiments [14]. Another example is vernier hyperacuity, first measured in 1892 by Wulffing [15]. The subject is presented with a pair of parallel vertical lines, one above the other, and offset horizontally by a small spatial displacement δx. Westheimer and McKee found that subjects could detect such a displacement down to about 5" of arc [16].

It is particularly interesting that the limits of resolution in hyperacuity experiments are much less than the spatial resolution of the photoreceptors in the retina; the spacing between foveal receptors is about 25" to 30" of arc. The implication is that at some level in the visual system, there is a finer-grain representation of at least part of the visual field than exists at the retinal level. This was in fact suggested by Barlow, [17]. In addition, it was suggested that the structure of layer 4Cβ of the striate cortex makes it a good candidate as a site where such a reconstruction could take place. Layer 4C is the main "input" to the striate cortex; in monkeys, the X or sustained-response cells, which are the predominant type of retinal ganglion cell, project to layer 4Cβ, via the lateral geniculate nuclei (LGN). It has been observed that that there are a relatively large number of cells in this layer in proportion to the number of incoming fibers. In particular, Crick et al. [18], as well as Barlow, used neuroanatomic data to estimate that there are one to two orders of magnitude more cells than incoming fibers, and suggested that incoming retinal data could be interpolated as it entered the visual cortex.
The preceding sentence was worded carefully; what was not said was that samples of the image intensity function that are being interpolated in layer $4C\beta$, but rather whatever data the retina sends back to the cortex. There is a fair amount of evidence, and a supporting computational theory, to suggest that the retina does not send back raw intensity values, but rather encodes local spatial changes in intensity; see Marr [19], and Marr and Poggio [20]. Motivation for this approach comes in part from the observation that intensity changes are important in that they often signal significant physical changes in one's field of view, especially physical edges of objects. Computational goals aside, such a sampling scheme is certainly valid in the light of Shannon's remarks; if we have enough independent numbers, ("2JW" in the previous context), then we have captured the information in the original function, and manipulation of these sample values is totally equivalent to manipulating the original function. Within this framework, it has been suggested that the information in the function is captured in the zero-crossings of the the function, after smoothing with bandlimited filters of different spatial scales [4,21,22]. This concept was initially supported by results of Logan [23], and more recently advanced by the results of Curtis [24], who showed that most two-dimensional bandlimited functions can be recovered from their zero-crossings. Thus whereas Barlow [17] spoke of reconstructing the retinal image, Crick et al. presented a more precise hypothesis. It was asserted that the cells in layer $4C\beta$ signal a fine-grain reconstruction of a band-pass filtered retinal image, with the goal of preserving the location of the zero-crossings (see also Marr and Hildreth [4]).

Now, the interpolation concepts introduced earlier considered only the case of continuous interpolation, i.e. producing a continuous result from discrete samples. However, for the case of hyperacuity, we are considering discrete interpolation, that is, interpolating one discrete function to produce another. Thus, we must consider the relevance of the continuous-domain results to these discrete-domain problems. It should be made clear that the continuous-domain results are completely transferable. The reason is that discrete interpolation can be viewed as a two-step process; interpolation to a continuous result, followed by resampling the continuous function in a different manner. The sampling theorem tells us under what conditions we can expect the resampled function to capture the information in the continuous function in this formulation, and thus in the original sampled function. Of course, in light of Shannon's remarks, the new set of samples will contain no more information than the original set. However, the new sample values may be better suited for a particular task. For example, in the case of
hyperacuity, the resampling would be at a higher sampling resolution, and these can then be used to locate image features with more precision, though not with more accuracy, than would be obtainable from the lower-resolution samples.

1.6 Temporal Sampling in Visual Systems

Now, we are ready begin focusing our discussion on the question implicitly asked earlier, namely, what is the visual system doing while one is watching a movie? It is clear that issues of temporal sampling and reconstruction are important here, and by now, we should be well equipped to address those issues.

We should start by clearly stating that there is no convincing evidence that the human visual system performs temporal sampling of the image intensity function. To be sure, there are several layers or stages of cells (retinal receptors, retinal ganglion cells, lateral geniculate nuclei, etc.) through which intensity information passes, and which are separated by temporal delays due to the propagation time of signals in neurons. However, there is nothing to suggest that these cells are acting in synchrony, to acquire and process image intensities in temporal "snapshots". In addition, there would seem to be no compelling reason to do so. Of course, ultimately, the measurement and processing of intensity information must be a discrete process; fundamentally because the emission and capture of photons are discrete random processes, but also because of the firing of many neurons is also well-described as a discrete random process. However, we will maintain the position that the human visual system processes the temporal variations in the image intensity function continuously, at least on our time-scale of interest, rather than by acquiring and manipulating a set of temporal samples of that function.

However, we can more-or-less sidestep this whole issue by considering intensity functions which are temporally pre-sampled, i.e. those which consist of a series of frames which represent temporal samples of some continuous intensity function. Then the visual system is constrained to process the frames in (loose) temporal synchrony. The obvious example is the one already mentioned, namely a movie. In addition, there are other vision systems for which a discrete temporal representation of the image is a natural one, in particular, computer-based vision systems. Thus, for the theoretical development
which follows shortly, we will focus on issues surrounding temporal sampling and reconstruction, not because that is what the human visual system is doing, but because in some cases, the visual system must deal with temporally sampled images, and because other vision systems can deal only with such a representation.

1.7 Temporal Interpolation in the Visual System

The fact that humans perceive continuous motion when we view a movie suggests that the visual system performs something similar to temporal interpolation. In addition, one can heuristically deduce the mechanism. The visual pathways, like any physical transmission channel, are known to be bandlimited. By this we mean that the higher frequency components of inputs to such a channel are not passed, but rather attenuated by the channel. Physically, this arises because of the finite time it takes to push ions around, or in biological systems, to change the state of an electrically active cell such as a neuron. In the frequency domain, this means that the frequency response of such a channel has a characteristic low-pass filter or band-pass filter shape; above some cutoff frequency, the frequency response approaches zero. As discussed previously, any filter which has has such a frequency response will implicitly interpolate a sampled function to which it is applied, as long as the replicates of the original spectrum are all eliminated by the filter. Thus, we could jump to the conclusion that there is nothing magic about temporal interpolation in the visual system: the reason a movie appears to us to represent continuous motion is because the visual system, because of its bandlimited temporal frequency-response, low-pass filters this discrete input, and presents the higher visual centers with an interpolated, continuously changing image.

The fact that the visual system is temporal bandlimited is implied by the results of Bloch in 1885 [25]. He found that the apparent intensity of a light flash was a function of the product of the actual intensity and the duration of the flash, for durations up to about 100 milliseconds. From this it has been inferred that the visual system integrates its input over approximately that duration. Since integration is a bandlimited operation, this is one way to interpolate a sampled function. This result has been greatly expanded by measurements of the temporal and temporal-frequency characteristics of the visual system, and of their relation to spatial-frequency, for example by Campbell and Robson
[26], Wilson and Gieze [27], and Wilson and Bergen [28]. Although there appears to be more than one spatio-temporal frequency channel in the human visual system, it is seems safe to say that they are all bandlimited to less than about 30 cycles per second. More specific details are given in the references mentioned; see also Robson [29], and Kelly [30].

1.8 Evidence for Spatio-temporal Interpolation : Apparent Motion

The failings of the previous conclusion can be demonstrated with a simple experiment. The viewer is presented with a pair of lights, which are located at points A and B, separated by some distance D. The first light blinks on and then off, and following some time delay T_1, the second light does the same. After another time delay T_2, the cycle is repeated. The viewer's perception, for fixed D, is as follows. When the delay T_1 is zero, one sees of course two lights blinking in synchrony. As T_1 increases, one still sees two lights, now blinking slightly asynchronously. However, as T_1 increases further, one sees something more interesting; one begins to see one light, moving from point A to point B. This phenomenon is referred to as apparent motion, or sometimes the β- or φ-effect [31-33]. As the delay T_1 increases further, the apparent motion effect breaks down, and one sees two lights blinking alternately.

The surprising thing is not that one has the perception of a moving object; this would seem to be a reasonable feature from evolutionary arguments. However, more than just sensing this motion, one actually perceives light coming from locations between A and B; i.e. one sees light coming from a place where there is no physical light source. It seems unlikely that the higher visual centers, having decided that something is moving in the visual field, would expend the effort to create this illusion. Rather, we postulate that, as a result of some lower-level process, the retinal signal is transformed and looks the same, or nearly so, as it would if there actually was a single moving light source.

It is not clear at first glance how it is that the light can appear to be in a spot where it is not. One is tempted to say that this is a result of a temporal interpolation process; after all we have discrete data being converted to a smooth motion. In addition, if the lights blinks on and off at a fast enough rate, they appear to be continuously illuminated; this is consistent with the bandlimited nature of the visual
pathways, and can rightfully be described as temporal interpolation. However, we must stress that the simple temporal interpolation scheme presented above cannot be responsible. To convince ourselves, we envision a three-dimensional coordinate system, where two of the axes are spatial ($x$ and $y$) and the third is temporal ($t$). Temporal interpolation acts by smoothing the image in a direction parallel to the $t$-axis. But in the apparent motion phenomenon, there is smoothing in both spatial and temporal dimensions simultaneously. This cannot be achieved by either temporal or spatial interpolation alone, but a combination which we will refer to as spatio-temporal interpolation.

1.9 Acuity and Motion

There is another set of experimental results that indicates that spatio-temporal interpolation is incorporated in the human visual system. These experiments test acuity, and especially hyperacuity, for moving targets. Westheimer and McKee [16], presented subjects with a sequence of discrete verniers, each one separated from its neighbor, spatially by some $\Delta x$ and temporally by some $\Delta t$. Theoretically, such a sequence could be generated by regular temporal sampling of a continuously moving vernier, in particular, one that is moving at constant velocity. The experimental results showed that vernier hyperacuity is maintained as the target moves in the velocity range from 0° per (temporal) second up to about 4° per second, as long as the apparent motion phenomenon was present. This would be a less interesting result if the eye was able to track the moving pattern, however the presentations were short enough (150 msec) that eye pursuit could not initiated, and thus cannot be considered a factor here [34].

Perhaps the most remarkable experimental result is due to Burr [35]. Subjects were presented with a discrete sequence of “moving” vertical line segments, as in Westheimer’s experiments. However, in this case, the top and bottom line segments of each image were not offset spatially, but rather they are displayed with a small temporal offset $\delta t$, on the order of a few milliseconds, relative to each other. When the display parameters are adjusted to the range where apparent motion is perceived, as described above, something quite intriguing happens; the viewer sees a moving vernier. In addition, the illusory spatial displacement between the top and bottom segments is consistent in size with the velocity and temporal offset, i.e. $\delta x \approx v \delta t$ for some velocities. And further, the acuity demonstrated
in distinguishing left- from right-oriented “verniers” falls in the hyperacuity acuity range, as long as apparent motion is perceived.

It is clear, hopefully, that simple temporal interpolation is insufficient to account for these results. Because the top and bottom segments of the verniers were spatially aligned in these experiments, simple temporal interpolation would result in a continuous image in these segments remained aligned. Of course, as already discussed, the apparent motion of the vernier is also inconsistent with simple temporal interpolation. Again, we will assert that such a phenomenon requires simultaneous temporal and spatial interpolation, i.e. spatio-temporal interpolation.

1.10 Towards a Computational Theory: the Constant Velocity Assumption

The results of Burr and of Westheimer were confirmed and extended by Fahle and Poggio [36], and a computational theory which is consistent with the results is presented by those authors and by Poggio et al. [37]. In formulating this theory, the stimuli used by Westheimer are modeled as temporal samples of a continuous, moving vernier. The stimuli used by Burr are modeled as spatial samples of the same vernier; this would be equivalent to viewing the moving vernier through a piece of cardboard with a single vertical slit in it. In fact, the reconstruction of moving images as viewed through stationary slits has been reported, and is sometimes referred to as Park's effect [38-40]. Fahle and Poggio examined the Fourier transforms of the sampled stimuli, and considered possible interpolation schemes in light of these frequency domain representations as well as psychophysical results.

It is shown that temporal sampling, as in Westheimer's experiments [16], preserves spatial information in a way that makes it very easy to recover. Each temporal sample, or frame, contains enough spatial information to detect the vernier and determine the direction of the displacement; thus the reconstruction of a temporally sampled moving vernier can be very poor without affecting vernier acuity. Indeed, if one merely adds the frames together, one gets a very poor reconstruction, perhaps best described as a grating composed of verniers, from which one can analyze the spatial configuration of the verniers, and perform the discrimination task. On the other hand, the spatial sampling
encountered in Burr's experiments [35] does not preserve spatial information in this way; to be sure the information is captured in the samples, but it is not easy to recover as it is in the temporally sampled case. In this case, fairly good reconstruction is needed to recover the spatial information, and especially to achieve acuity in the hyperacuity range. The authors concur with our argument that simple spatial or temporal interpolation schemes are not sufficient by themselves to develop a computational theory which is consistent with known properties of the human visual system, but rather that a dependent interaction of the two, *i.e.* spatio-temporal interpolation, is required.

Fahle and Poggio [36] proposed a framework for understanding spatio-temporal interpolation based on a simple observation; in the physical world, over suitably small times and spatial regions, things moving in our visual field represent rigid-body motion at nearly constant velocity. This *constant velocity assumption* leads to some very interesting things in the frequency domain, which in turn lead to some surprising, though intuitively correct, results relating to sampling and interpolation.

In the next chapter, we will focus on the theoretical aspects of the constant velocity assumption, starting in the frequency domain, and show how the assumption can be used to design a linear filter which is tuned for objects moving at a constant velocity, *i.e.* a *velocity-tuned filter*. Then, we will explore the implications of the constant velocity assumption in the context of sampling and interpolation, and show how, using a velocity-tuned filter, one can reconstruct the continuous image depicting constant-velocity motion from a surprisingly small number of samples.

In parallel, we will investigate the space-time effects of the velocity-tuned filter. We mentioned before that the effect of a linear filter is embodied in the convolution operation, and in general, a three-dimensional filter would entail a three-dimensional convolution (two spatial and one temporal dimension), which is straightforward, but quite computationally expensive. Interestingly, as we will show, the velocity-tuned filter can be implemented, and thus *should* be, as a one-dimensional convolution in three-dimensional space, oriented in a direction which depends on the velocity to which the filter is tuned. Not only does this ease the computational strain, but it makes the space-time operation intuitive; it demonstrates that the velocity-tuned filter performs spatio-temporal interpolation by smoothing along one-dimensional constant-velocity lines in the space-time domain.
Now, these results will be based on the rather stringent assumptions that we have motion that can be described by the translation of a well-defined function, for all time and space, and that we know the velocity of the translation. Of course, in a real image-processing situation, these assumptions are prohibitively tight. So we will also concentrate on ways in which they can be relaxed. In particular, we will consider filters that are more broadly tuned to a range of velocities rather than to a single velocity, and also consider the effects of translation at non-constant velocities. In addition, we will consider the effects of constant-velocity translation over finite segments of time and space.

There is one more reality to face. As discussed previously, we can never determine sample values exactly, as these measurements are destined to be corrupted by noise. One approach would be to use statistical estimation theory to determine the "best estimate" for the actual sample values, and go from there. We have chosen another path, however. We take the stance that the addition of noise makes the spatio-temporal interpolation task an ill-posed problem, and deal with this situation in the appropriate way, which is to apply regularization theory to find an optimal interpolation filter. Those who are unfamiliar with this approach can look forward to a clear explanation in the next chapter. Estimation theorists may feel comforted to hear that there is evidence that the statistical approach and the regularization approach produce equivalent results. Pleasingly enough, the regularized solution that we put forth here, which is an approximation filter rather than an interpolation filter, turns out to be a velocity-tuned filter. Thus we can use the previous theoretical results to characterize the regularized filter, and at the same time use the regularization results to support the velocity-tuned filter approach.

The following chapters will discuss implementation issues as needed, and present experimental results to verify the operation of the spatio-temporal interpolation and approximation filters, and to examine their properties. Finally, new psychophysical results will be presented which verify a specific prediction that follows from the computational theory; the nature of the smoothing performed in velocity-tuned spatio-temporal interpolation and approximation filters implies that objects which are moving at constant velocity should be more detectable in the presence of noise than objects which are moving randomly. We shall see that this prediction is borne out experimentally.
2.1 Constant-Velocity Theorems

The theoretical development of velocity-tuned filters is based on the frequency domain description of images which represent motion of objects with constant velocity. Of course, in real images, we would never expect objects to move with constant velocity for all time. Indeed, things are even more complex since we might expect the shape of objects to change due for example to rotation or occlusion. However, the constant-velocity case is a good theoretical starting point, and will lay the groundwork for considering the effect of more arbitrary motion, in Section 2.5.

Therefore, we now start by considering the implications of a rather simple assumption, namely that we have a time-varying image which represents constant-velocity translation of a rigid body. This we call the constant-velocity assumption. Mathematically, this implies that the variables of the image function are not all independent, but rather are related to each other through the velocity parameters.

To create a two-dimensional constant-velocity function, \( g(x,t) \), we translate one-dimensional function at constant velocity, \( v \), thus:

\[
g(x,t) = f(x - vt)
\]
An example is shown in Figure 2.1.1. Now, as shown in Appendix C, constant-velocity functions have a rather interesting Fourier transform. In particular, in this two-dimensional case:

**Theorem 2.1.1**

If a two-dimensional function, \( g(x, t) \) can be expressed as a constant-velocity translation of a one-dimensional function \( f(x) \), so that
\[
g(x, t) = f(x - vt),
\]
then the Fourier transform of \( g(x, t) \) is given by:

\[
G(\omega_x, \omega_t) = F(\omega_x) \, \delta_l(\omega_x v + \omega_t)
\]

\[
= F(\frac{-\omega_t}{v}) \, \delta_l(\omega_x v + \omega_t)
\]

We use \( \delta_l(z) \) to represent an impulse line. As described in Appendix A, this is a function for which, at each point along the line described by \( z = 0 \), one finds a unit impulse. The dimensionality of the impulse is given by the number of arguments in the function. Thus, the Fourier transform of a two-dimensional constant-velocity function has the form of a line of one-dimensional impulses. This is a fairly unusual thing to find in a two-dimensional space, but as will be shown in the next section, leads to intuitive results and a simple strategy for implementing filters based on the constant-velocity assumption.

A typical spectrum of a two-dimensional constant-velocity function is shown in Figure 2.1.2. Of particular importance is that the support of the Fourier transform of a two-dimensional constant-velocity function, *i.e.* the region of frequency space where the Fourier transform is non-zero, is a line in the two-dimensional frequency plane.
Figure 2.1.1 Constructing a Two-dimensional Constant-Velocity Function
Spectrum of One-Dimensional Function (Magnitude)

2D Spectrum of Constant-Velocity Function (Magnitude)

Figure 2.1.2 Constant-Velocity Functions in the Frequency Domain
Section 2.1.1

Constant-Velocity Theorems

Extension of the above discussion to three dimensions is straightforward. To create a three-dimensional constant-velocity function, \( g(x, y, t) \), we translate two-dimensional function, (which could represent a two-dimensional image, for example), at constant velocity in the \( x \) and \( y \) dimensions. Thus:

\[
g(x, y, t) = f(x - v_xt, y - v_yt)
\]

Such a function is illustrated in Figure 2.1.3. The three-dimensional Fourier transform of \( g(x, y, t) \) has a very similar form to the two-dimensional case, as shown in Appendix C:

\[
\textbf{Theorem 2.1.2}
\]

If a three-dimensional function, \( g(x, y, t) \) can be expressed as a constant-velocity translation of a two-dimensional function, \( f(x, y) \), so that \( g(x, y, t) = f(x - v_xt, y - v_yt) \), then the Fourier transform of \( g(x, y, t) \) is given by:

\[
G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) \delta_p(\omega_x v_x + \omega_y v_y + \omega_t)
\]

Here, \( \delta_p(x) \) is an impulse plane. By analogy to the impulse line, the impulse plane is a function for which, at every point on the plane which is described by \( z = 0 \), we find a unit impulse. Again, the number of arguments expresses the dimensionality of the impulses. In this particular case then, the Fourier transform for the velocity-tuned filter is a plane of one-dimensional impulses, which goes through the origin of three-dimensional frequency space, and whose orientation is governed by the \( x \) and \( y \) components of the velocity of the two-dimensional function.

From the two-dimensional and three-dimensional results, we expect extension to \( n \)-dimensional signals would be straightforward; however we will only have use for two- and three-dimensional results.
Figure 2.1.3  A Three-dimensional Constant-Velocity Function
2.1.1 Implications of the Constant-Velocity Theorems

As pointed out by Fahle and Poggio [36], and Poggio et al. [37], an important implication of the constant-velocity theorems is that one can sample constant-velocity functions, without aliasing, at rates considerably lower than one would expect for an arbitrary function. To see that this is true, we need only examine the situation in the frequency domain.

As discussed in Appendix B, sampling a function results in the replication of the Fourier transform, or frequency spectrum of the function along the frequency axis which corresponds to the variable in which the function is sampled. As further discussed, the original function can be recovered exactly from its samples if and only if the spectral replicates do not overlap. For arbitrary functions, it is necessary and sufficient that the function be sampled at a rate greater that twice the frequency of the highest frequency component of the function, a quantity known as the Nyquist rate, in order that the spectral replicates not overlap.

Thus, as depicted in Figure 2.1.4, sampling a two-dimensional constant-velocity function in time results in replication of its frequency spectrum along the temporal-frequency axis. Now, if the one-dimensional function \( f(x) \) from which we generated the constant-velocity function, is bandlimited in one-dimensional frequency space to \( |f| \leq W \), then the constant-velocity function \( g(x, t) \) is bandlimited in two-dimensional frequency space to \( |f_x| \leq W \) and \( |f_t| \leq vW \). Thus, if the sampling rate exceeds the Nyquist rate, we can recover the original function in a standard fashion by using a low-pass filter:

\[
H(f_x, f_t) = \begin{cases} 
1, & \text{if } |f_x| < W \text{ and } |f_t| < vW \\
0, & \text{otherwise.}
\end{cases}
\]

This type of filter is shown in Figure 2.1.4. If the sampling rate falls below the Nyquist Rate, then this approach no longer works. From the frequency domain depiction, there are three possible ways to proceed. As show in Figure 2.1.5, one is to use a low-pass filter which recovers the original spectrum, but also passes part of the neighboring spectral replicates, thereby adding unwanted frequency components to the original function. A second approach is to use a low-pass filter that is just small
Figure 2.1.4 Sampling and Reconstruction of Constant-Velocity Functions
Parts of Spectral Replicates Added to Original Spectrum

Original Spectrum is Recovered, but is Low-Pass Filtered

Figure 2.1.5 Lowpass-Filter Reconstruction Fails at Lower Sampling Rates
enough to exclude the spectral replicates; this will also result in low-pass filtering the original function, but this may be acceptable in some situations.

These approaches seem simple and straightforward enough, but they do not exploit the geometry of this situation. In particular, there is a significant difference here, relative to the general case. As explained in Appendix B, when the sampling rate of an arbitrary function falls below the Nyquist rate, the spectral replicates are irrevocably merged, so that the original spectrum cannot be recovered. This phenomenon is called *aliasing*, presumably because the resulting sampled function is indistinguishable from the result of sampling some other, bandlimited function at its Nyquist rate. In the case of a constant-velocity function however, even at low sampling rates, we don't get aliasing. The spectral replicates are still distinct, but they can no longer be recovered by a simple low-pass filter scheme. However, if the reconstruction filter is "oriented" along the direction of the support of the original spectrum in the spatio-temporal frequency plane, we can sample at rates less than the equivalent Nyquist rate, and still expect to be able to reconstruct the original signal. Indeed, because the support of the original spectrum is a line, there should never be any overlap between the spectral replicates, as long as the sampling rate is non-zero. Thus, in the limit of this argument, if the reconstruction filter has a support that is a line which is colinear with the support of the original spectrum, then the sampling rate can drop to any non-zero value, and reconstruction will still be possible. In addition, this holds true irregardless of whether or not the original signal was bandlimited.

It might seem rather surprising that we can drop sampling rate so low, and still expect to be able to recover the original function. However, a moment's thought will reveal the intuitiveness of this result. Having made the assumption that a function is composed of some other function moving at a constant velocity, we only need one sample to completely specify the function for all time. By analogy, consider a line in the \((x, y)\) plane; knowledge of the slope of the line and knowing that the line goes through some point \((x_0, y_0)\), is enough information to construct the line for all \((x, y)\). Similarly, knowing that a function is a constant-velocity function with velocity \(v\), and having a sample of that function at some time \(t_0\) is provides enough information to completely reconstruct the entire function. This is what the frequency domain argument tells us; as long as we have one sample, the sampling rate can get arbitrarily low.
Section 2.2.0

Velocity-Tuned Filters

It should be noted that the same results apply to the case of spatial sampling of a constant-velocity function, and in fact to the case when the function is sampled in both space and time. Spatial sampling results in replication of the original spectrum along the spatial frequency axis; if the original function is bandlimited, and if it is sampled at a sufficiently high rate, then a standard low-pass filter can be used to reconstruct the original function. However, the geometry of the support of the original spectrum guarantees that there will never be any spectral overlap, so that reconstruction of the original function will always be possible as long as the sampling rate is non-zero, if we use a filter that exploits this geometry. This is illustrated in Figure 2.1.6.

A filter which is oriented in frequency space, would be "tuned" for reconstruction of constant-velocity functions over some range of velocities. We will refer to such a filter as a velocity-tuned filter, and will spend considerable time in the next few sections examining ways of constructing these filters and looking at their properties.

2.2 Velocity-Tuned Filters

In this section, we consider how we may exploit the properties of the Fourier transform of constant-velocity functions, as exposed in the previous section, in order to construct velocity-tuned filters. We will then examine the effect such a filter has on several classes of input functions, followed by an examination of the use of such a filter as an interpolation filter for reconstructing sampled functions. In the following text, the phrase "velocity of the filter" will be used freely to mean the velocity to which the filter is tuned, rather than any other more confusing meaning; this quantity will be represented by \( v_f \). Similarly, we will use the term "velocity of the input" to mean the velocity of a constant-velocity function; this will be denoted as \( v_i \).
Figure 2.1.6 Reconstruction of Constant-Velocity Function with Velocity-Tuned Filter
2.2.1 Two-dimensional Filters

We start with a simple, two-dimensional case. We wish to construct a filter which has unity gain over a small region around the support of the constant-velocity function in the frequency plane. We cannot use a function that has unity gain only over the line of the support; since the "volume" under such a function is zero, it would be meaningless to talk about its inverse Fourier transform. Instead, we construct a filter by translating a one-dimensional function along the line $\omega_t + v f \omega_z = 0$ in the frequency plane. We will use the term filter generator function to mean a function which is used to generate a filter function in this, or a similar manner. Since the support of a constant-velocity function always goes through the origin, the support of the filter generator function will be cover some region around the origin, i.e. it will pass relatively low frequencies and suppress relatively higher frequencies. Thus the filter generator function will always be a low-pass filter.

Now, we make use of a theorem which is proven in Appendix C :

**Theorem 2.2.1**

If a two-dimensional function, $g(x, t)$, has a Fourier transform which has the form of a constant-velocity translation of a one-dimensional function, $F(\omega_t)$, so that $G(\omega_z, \omega_t) = F(v \omega_z + \omega_t)$, then the function $g(x, t)$ is given by :

$$g(x, t) = f(t) \delta_t (x - vt)$$

$$= f \left( \frac{x}{v} \right) \delta_t (x - vt)$$

In other words, the impulse response of this filter, that is, the inverse Fourier transform of $H(\omega_z, \omega_t)$, is a line of one-dimensional impulses, whose coefficients can be found from the inverse Fourier transform of the one-dimensional filter generator function.
As an example of this approach, consider a velocity-tuned filter produced by a rectangular-pulse filter generator function. Such a filter is shown in Figure 2.2.1.

\[
F(\omega_t) = \begin{cases} 
1, & -B < \omega_t \leq B \\
0, & \text{else}
\end{cases}
\]

\[
H(\omega_x, \omega_t) = F(v \omega_x + \omega_t)
\]

Here, we define \( B \) as the \textit{temporal bandwidth} of the filter generator function. The inverse Fourier transform of the filter generator function is the well known \textit{sinc} function:

\[
f(t) = \frac{\sin 2\pi Bt}{2\pi Bt}
\]

So the two-dimensional impulse response of the velocity-tuned filter becomes:

\[
h(x, t) = \frac{\sin 2\pi B(x - vt)}{2\pi B(x - vt)} \delta_t(x - vt)
\]

This function is depicted in Figure 2.2.2.

### 2.2.2 Space-time Domain Description of Two-dimensional Filter

Now, we wish to characterize the velocity-tuned filtering operation in the space-time domain. As explained in Appendix A, the output of this linear filter can be described as the two-dimensional convolution of the input function with the impulse response of the filter. The result of a two-dimensional convolution with a function comprised of one-dimensional impulses may not intuitive.
Figure 2.2.1 A Rectangular-Pulse Velocity-Tuned Filter
Figure 2.2.2 Rectangular Velocity-Tuned Filter in Space-Time
However, we have shown, in Appendix D:

**Theorem 2.2.2**

Consider a two-dimensional filter with the form:

\[ g(x, t) = f(t) \delta_l(x - vt) \]

where \( \delta_l(x - vt) \) is a line of one-dimensional impulses, as described in Appendix A, traversing the \((x, t)\) plane at constant velocity. Then, convolving an arbitrary two-dimensional image, \( s(x, t) \) with \( g(x, t) \) has the effect of convolving each one-dimensional, constant-velocity "slice" of the image, \( s(x_0 + vt, t) \), by \( f(t) \).

In other words, we can think of this filter as performing the following operation on a two-dimensional input function. A one-dimensional slice of the function is extracted, consisting of the value of the input function along the line \( x = x_0 + vt \). Then the corresponding slice of the output is formed by convolving the slice of the input with the one-dimensional function, \( f(t) \). This operation is depicted graphically in Figure 2.2.3.

### 2.2.3 Effects of the Two-dimensional Filter

Now we consider the actions of this filter on various types of inputs. The filtering operation is expressed in the frequency domain by the multiplication of the Fourier transforms of the input and of the impulse response of the filter; this product is then the Fourier transform of the output function.
Figure 2.2.3 The Effect of Velocity-Tuned Filter Operation
First, let us see what happens when the input function is a constant-velocity function. From above results, we can write the Fourier transform of the output as:

\[
Y(\omega_x, \omega_t) = \frac{F(\omega_x) \delta_t (\omega_x v_i + \omega_t)}{\text{input}} \frac{H(\omega_x v_f + \omega_t)}{\text{filter}}
\]

Since the impulse line is zero when its argument is non-zero, we can simplify the argument of \(H\), by setting \(\omega_x v_i + \omega_t = 0\), thus:

\[
Y(\omega_x, \omega_t) = F(\omega_x) H(\omega_x (v_f - v_i)) \delta_t (\omega_x v_i + \omega_t) = Z(\omega_x) \delta_t (\omega_x v_i + \omega_t)
\]

where:

\[
Z(\omega_x) = F(\omega_x) H(\omega_x (v_f - v_i))
\]

Now, by comparing this result with the two-dimensional constant-velocity theorem, Theorem 2.1.1, we conclude that the inverse Fourier transform of \(Y(\omega_x, \omega_t)\) must be a constant-velocity function, of velocity \(v_i\), i.e. the same velocity as the input. We can find an explicit expression for \(y(x, t)\) by applying the constant-velocity theorem:

\[
y(x, t) = z(x - vt)
\]

where \(z(x)\) is the inverse Fourier transform of \(Z(\omega_x)\). To find \(z(x)\), we apply the frequency multiplication and the frequency scaling properties of the Fourier transform, which are described in Appendix A. The result is:

\[
z(x) = f(x) \ast \frac{1}{|v_i - v_f|} h \left( \frac{x}{v_i - v_f} \right)
\]
Section 2.2.3

Velocity-Tuned Filters

In words, \( z(x) \) is the one-dimensional convolution of \( f(x) \), i.e. the one-dimensional function which is being translated at constant velocity, and a space-scaled \( h(x) \). We note that \( h(x) \) comes from changing the variable of \( h(t) \), i.e. the inverse Fourier transform of the filter generator function. The fact that we are able to do this reflects the interdependence of spatial and temporal variables under the constant-velocity assumption; space and time are strictly related through the velocity parameter. Analytically, it is expressed by the various delta functions in previous (and subsequent) relations; in this specific case, it is expressed in the term \( \delta_t (\omega_x v_t + \omega_z) \), which allowed us to make the substitution \( \omega_t = -\omega_x v_t \).

Now the filter generator function is always a low-pass filter, which has the effect of smoothing or smearing the input. As the input and filter velocities become closer, the effect of the space-scaling is to spatially compress \( h(x) \), concentrating it around \( x = 0 \), which decreases its ability to smooth. Conversely, as the velocities become more disparate, \( h(x) \) gets stretched out, which makes it a more powerful smoothing filter. Thus, \( z(x) \) will be a smoothed version of \( f(x) \), and the amount of smoothing depends not only on the smoothness of \( h(x) \), but on the difference between the velocities as well. In summary, the output of the filter is a constant-velocity function of the same velocity as the input, but the function which is being translated has been smoothed by the generator function of the velocity-tuned filter.

We see from this result, and from Figure 2.2.4, that there are a few cases to consider. If the input signal is bandlimited, and if \( H (\omega_x (v_f - v_i)) \) is almost constant over the support of the input spectrum, then the input is passed almost unchanged. In order to get no change at all, \( H \) must clearly be constant over the support of the \( F(\omega_x) \); this would be true if the one-dimensional filter generator function is a rectangular pulse of sufficient bandwidth, for example. If these conditions are not met, the higher frequency components of \( F(\omega_x) \) will be filtered out. This phenomenon obviously gets more pronounced as (1) the bandwidth of \( H(\omega_x) \) gets smaller, (2) the velocity of the input becomes more disparate from the velocity of the filter (3) the bandwidth of the input increases.
Input is Passed Unchanged

Input is Low-pass Filtered

Figure 2.2.4 Effect of Velocity-Tuned Filter on Constant-Velocity Inputs
In the special case where the velocity of the input equals the velocity of the filter, the Fourier transform of the filter output becomes:

\[ Y(\omega_x, \omega_t) = F(\omega_x) H(0) \delta_t (\omega_x v_t + \omega_t) \]

In other words, the output of the filter equals the input, scaled by the constant, \( H(0) \). This is consistent with the space-time domain argument above. As \( \nu_f - \nu_i \) approaches zero, the second term of the convolution in Equation 2.2.2 behaves as an impulse function, which means that \( z(x) \) will equal \( f(x) \). Then the output of the filter will be a constant-velocity translation of the same one-dimensional function as the input, and at the same velocity as the input; thus the output and input will be identical.

Finally, we consider the case that the input is not a constant-velocity function. From the frequency domain, as depicted in Figure 2.2.5, we see that the output of the filter will tend to look “almost like” a constant-velocity function; in particular, the spectrum of the output will have a support which is oriented along a constant-velocity line in the frequency plane. The spectrum will, in general, differ from that of a constant-velocity function because the one-dimensional slices are not guaranteed to be identical, as they are in the spectrum of a constant-velocity function. Moreover, if the spectrum of the input is strongly oriented in the frequency plane, as in the preceding case, the “orienting effect” of the velocity-tuned filter may be obscured. The effect of this filter on arbitrary inputs will be more exactly described subsequently.

Much insight into the effect of the velocity-tuned filter is gained by considering the situation in the space-time domain. We saw that the effect of the filter is to convolve one-dimensional, constant-velocity “slices” of the input with the inverse Fourier transform of the one-dimensional function which was used to generate the two-dimensional filter. In one particular case, the generator function is a rectangular pulse, so the the function which we convolve with is a sinc function; we will consider later other reasonable functions. Since all constant-velocity lines in the frequency plane go through the origin, the support of the velocity-tuned filter must include the origin. As a consequence, a velocity-tuned filter will be some sort of low-pass filter. The inverse Fourier transform of a low-pass filter is going to be
Figure 2.2.5 Effect of Velocity-Tuned Filter on Arbitrary Inputs
some sort of interpolating function, which will have tendency to "smear out" functions with which it is convolved; the sinc function previously discussed is a good example of this. Thus the space-time effect of the velocity-tuned filter is to smear the input image along lines of constant-velocity, namely the velocity of the filter.

Thus, when the input function is a constant-velocity function with the same velocity as the filter, it is passed unchanged because, roughly speaking, it is being smeared out onto itself. An appropriate analogy is a painter, who always paints with a straight stroke, and with a constant stroke orientation. If she paints over a stripe on the wall which is smooth, straight, and which has the same orientation as her own stroke, the stripe will be appear unchanged by the painter's action (assuming the same color paint is used).

In other cases, the input function will tend to be smeared by the filter along the lines which are parallel to the constant-velocity line of the space-time domain filter, namely \( x = v_f t \); the smearing is due to the convolution by a low-pass filter function such as the sinc. Thus, for inputs which are not oriented along lines parallel to the constant-velocity line of the filter, convolution with a velocity-tuned filter will smear the input along those lines. Again, by analogy, if the painter paints over an oriented stripe on the wall, using fixed-length strokes of orientation different than that of the stripe, the tendency will be to smear the stripe in the direction of the stroke. The same argument clearly applies to arbitrary, non-oriented patterns on the wall.

There is a seeming contradiction in these statements. In the space-time domain, we say that any input which is not a constant-velocity function of the same velocity as the filter will get smeared. Whereas, using frequency domain arguments, we claim that any input whose bandwidth is less than that of the velocity-tuned filter may be passed unchanged by the filter. Since the space-time filter has the form of a sinc function, how can it be that the input function is not only not smeared by this filter, but is left unchanged as well? The answer is that is a property of bandlimited functions that if they are convolved with any function of the form \( \sin(ax)/(ax) \), when \( a \) is larger than highest frequency component of the input, then the input function is left unchanged. This unintuitive result can be demonstrated by evaluating the appropriate convolution integral, but it is much more readily
seen by considering the situation in the frequency domain; the Fourier transform of the sinc function is a rectangular pulse, which passes unchanged any function that is sufficiently bandlimited.

2.2.4 Effective Velocity Bandwidth of the Two-dimensional Filter

It is clear from this discussion that it is reasonable to talk about the bandwidth of a velocity-tuned filter. For the following discussion, let us assume that in our construction of the velocity-tuned filter, we use a one-dimensional rectangular-pulse filter generator function, whose temporal bandwidth was $B$.

To characterize the bandwidth of the velocity-tuned filter, let us consider the situation in Figure 2.2.6, where we have a the support of the spectrum of a bandlimited constant-velocity function superposed on the support of a velocity-tuned filter. Here we delimit the support of the filter by its bandwidth. This approximation may be very good or very bad; it is only exactly correct when the filter generator function is a rectangular pulse.

From this figure, we can define the relative temporal bandwidth, $B_t$, of the filter as the temporal bandwidth relative to the constant-velocity line of the filter; i.e. this is the bandwidth of a slice through the support of the velocity-tuned filter in the temporal frequency dimension. This is the same as the bandwidth of the one-dimensional filter generator function, namely $B$. We can also define the relative spatial bandwidth, $B_s$ of the filter as the spatial bandwidth, relative to the constant-velocity line. This is the bandwidth of a slice of the two-dimensional Fourier transform in the spatial frequency dimension. We can derive this by looking at the intercepts of the support of the constant-velocity function with the temporal frequency axis; thus, from the figure, we get $-B/\nu_f$. Finally, we define the velocity bandwidth as the width of the support of the velocity-tuned filter, perpendicular to the constant-velocity line. From geometry, it is clear that this is related to the relative spatial and temporal bandwidths by Pythagoras' Theorem. In this case, we would get $B\sqrt{1+(1/\nu_f)}$.

These quantities can all be useful in describing a velocity-tuned filter, but they do not tell us directly how much a given input will be altered upon passing through this filter. We wish now to derive
Figure 2.2.6 Critical Point for Determining Velocity Bandwidth
a quantity that embodies this information. With reference to Figure 2.2.6, it is clear from the geometry of the situation that the input function will be passed nearly unchanged by the filter if and only if the bandwidth of the input function is small enough that its spectrum can be nearly all contained within that part of the filter support where the filter is flat, or nearly so.

From Figure 2.2.6, we can write the equations for the support of the input spectrum, and of the lower and upper limits of the “flat region” of the filter. Here we assume that the filter is nearly flat for frequencies within the filter bandwidth.

\[
\begin{align*}
\text{input support:} & \quad f_i^l(f) = (-v_i) f \\
\text{upper limit:} & \quad f_i^u(f) = (-v_f) f + B_t \\
\text{lower limit:} & \quad f_i^l(f) = (-v_f) f - B_t
\end{align*}
\]

where \( B_t \) is the relative temporal bandwidth as defined above. The critical point occurs at \( f = W \), where \( W \) is the (spatial) bandwidth of the input function. At this point, the following conditions must hold in order for the input to be nearly unchanged:

\[
\begin{align*}
& f_i^u(W) > f_i^l(W) \\
& f_i^l(W) < f_i^u(W)
\end{align*}
\]

From these, we can derive the following conditions on the velocity of the input:

\[
v_f - \frac{B_t}{W} < v_i < v_f + \frac{B_t}{W}
\]

2.2.4

We can restate this result as follows. If the input is a constant-velocity function, spatially bandlimited to \( |f| < W \), and if the filter generator function of the velocity-tuned filter is temporally bandlimited to \( B_t \) and if the velocity of the input satisfies \( v_i = v_f \pm B_t/W \), then the function will not be changed much by passing through this filter. For obvious reasons then, we define the quantity \( B_t/W \) as the effective velocity bandwidth of the filter.
If the input is a constant-velocity function for which this condition is not met, then the output of the filter will still be a constant-velocity function, and its velocity will be unchanged, but each one-dimensional slice will be low-pass filtered, or smeared, relative to the input. Note that this is in contrast to a simple narrow-band filter, for which, if the input frequency is outside the passband of the filter, greatly attenuates the function.

Figure 2.2.7 shows that any arbitrary function may be passed almost unchanged by a velocity-tuned filter, if the function is suitably bandlimited in both space and time, and if the velocity-tuned filter is sufficiently flat over the support of the input spectrum. In this case, if the function is bandlimited to $W_z$ in spatial frequency, and $W_t$ in temporal frequency, then a necessary and sufficient condition for this to be true is that the upper limit of the filter support pass above the upper-right corner of the input spectrum. Thus, using the same notation as before:

$$f_t^*(W_z) > W_t$$
$$f_t^*(-W_z) > W_t$$

From the geometry, we can easily prove that this will be true when the following equivalent conditions holds:

$$B_t > W_t \pm v_f W_z$$  \hspace{1cm} 2.2.5

or equivalently:

$$B_t > W_t + |v_f W_z|$$

2.2.5 Resolution Tradeoffs in Two-dimensional Case

It is clear from the preceding section that, in designing and using a velocity-tuned filter, there is a fundamental tradeoff to be made. The smaller the velocity bandwidth requirement becomes, the smaller the bandwidth of the filter generator function. Then the inverse Fourier transform of the filter
Figure 2.2.7 Critical Point for Passing Arbitrary Function
generator function becomes longer. Since it is this function that is convolved with constant-velocity slices of the input, as described above, any function which falls outside the bandwidth of the velocity-tuned filter, will experience greater smearing. In addition, the conditions that a function must meet to avoid smearing will become more stringent. In the limit, as the velocity bandwidth goes to zero, only constant-velocity functions with the same velocity as the filter will be passed unchanged. Meanwhile, the one-dimensional convolution filter approaches a straight line. Thus, any input which is not a constant-velocity function with the same velocity as the filter is essentially smeared out indefinitely along the direction of the constant-velocity line of the filter.

2.2.6 Three dimensional Velocity-tuned Filter

In general, the previous results are extendable to three-dimensions without surprise. A three-dimensional image depicting two-dimensional rigid-body translation would have the form:

\[ g(x, y, t) = f(x - v_xt, y - v_yt) \]

As shown in Appendix C.

**Theorem 2.2.3**

If a three-dimensional function, \( g(x, y, t) \) can be expressed as a constant-velocity translation of a two-dimensional function, \( f(x, y) \), so that \( g(x, y, t) = f(x - v_xt, y - v_yt) \), then the Fourier transform of \( g(x, y, t) \) is given by:

\[
G(\omega_x, \omega_y, \omega_z) = F(\omega_x, \omega_y) \delta_p(\omega_x v_x + \omega_y v_y + \omega_z)
\]

Thus, \( G(\omega_x, \omega_y, \omega_z) \) has the form of a plane of one-dimensional impulses in three-dimensional \((\omega_x, \omega_y, \omega_z)\) space; specifically, the support of the Fourier transform is a plane described by:

\[ \omega_x v_x + \omega_y v_y + \omega_z = 0 \]

which goes through the origin. The coefficient of each of the impulses is given by the value of \( F(\omega_x, \omega_y) \), i.e. the Fourier transform of the zero-velocity function, at that point.
To build a filter tuned to such a function, we take the same basic approach as in the two-dimensional case. We start with a one-dimensional filter generator function, \( F(\omega_t) \), and translate it through frequency space along the constant-velocity line of the function described above; thus:

\[
H(\omega_x, \omega_y, \omega_t) = F(\omega_x v_x + \omega_y v_y + \omega_t). 
\]

The effect of this translation is that a plane, which is parallel to the plane of impulses described above, passes through each point in the support of \( F(\omega_t) \), which lies along the \( \omega_t \) axis. Furthermore, the value of \( F(\omega_t) \) at each point in its support is given to each point in the plane which passes through it. Thus, we generate a three-dimensional neighborhood which lies symmetrically with respect to the plane of impulses; any slice through the support in a direction parallel to the \( \omega_t \) axis would give a profile of values that would be identical to those found along the \( \omega_t \) axis, just shifted in the \( \omega_t \) direction. In effect, we have created a “sandwich”, as shown in Figure 2.2.8; the impulse plane that we are trying to recover lies in the middle of the sandwich. Any one-dimensional slice perpendicular to the surface of the sandwich yields the one-dimensional filter generator function. As in the two-dimensional case, the support of a three-dimensional constant-velocity function goes through the origin of frequency space. Thus the support of the filter generator function will be some region around the origin, making it a low-pass filter.

Now, we refer to a theorem proven in Appendix C:

**Theorem 2.2.4**

If a three-dimensional function, \( g(x, y, t) \) has a Fourier transform which has the form of a constant-velocity translation of a one-dimensional function, \( F(\omega_t) \), so that \( G(\omega_x, \omega_y, \omega_t) = F(\omega_x v_x + \omega_y v_y + \omega_t) \), then the function \( g(x, y, t) \) is given by:

\[
g(x, y, t) = f(t) \delta_t(x - v_x t, y - v_y t) 
\]

In other words, the impulse response of the velocity-tuned filter has the form of a line of two-dimensional impulses in three-dimensional \((x, y, t)\) space; this is in contrast to previous cases, where
Figure 2.2.8 Support of Three-Dimensional Velocity-Tuned Filter
(first quadrant has been cut away)
the impulses were one-dimensional. The coefficient of each impulse is given by the value of \( f(t) \), i.e. the inverse Fourier transform of the zero-velocity function, at that point.

As in the two-dimensional case, we use as an example the case where the one-dimensional filter generator function is a rectangular pulse. Thus:

\[
F(\omega_t) = \begin{cases} 
1, & -B < \omega_t \leq B \\
0, & \text{else}
\end{cases}
\]

\[H(\omega_x, \omega_y, \omega t) = F(\omega_x v_x + \omega_y v_y + \omega t)\]

The inverse Fourier transform of \( F(\omega_t) \) is still a sinc function, so:

\[
f(t) = \frac{\sin 2\pi B t}{2\pi B t}
\]

\[h(x, y, t) = \frac{\sin 2\pi B (x + v_xt, y + v_yt)}{2\pi B (x + v_xt, y + v_yt)} \delta_i (x + v_xt, y + v_yt)\]

2.2.7 Space-time Domain Description of Three-dimensional Filter

Now, when we filter an arbitrary, three-dimensional function with a filter of this type, the filtering operation can be expressed as a three-dimensional convolution of the impulse response, \( h(x, y, t) \), with the input function. Here we have two-dimensional impulses in the impulse response, whereas in the two-dimensional case the impulses were one-dimensional. However, the three-dimensional convolution contains a triple integral, rather than a double, so we get a very similar result to the two-dimensional case, as proven in Appendix D:
Consider a three-dimensional filter with the form:

\[ g(x, y, t) = f(t) \delta_l(x - v_xt, y - v_yt) \]

where \( \delta_l(x - v_xt, y - v_yt) \) is a line of two-dimensional impulses, as described in Appendix A, traversing \( (x, y, t) \) space at constant velocity. Then convolving an arbitrary three-dimensional image, \( s(x, y, t) \), with \( g(x, y, t) \) has the effect of convolving each one-dimensional, constant-velocity "slice" of the image, \( s(x_0 + v_xt, y_0 + v_yt, t) \), by \( f(t) \).

Thus, in direct analogy to the two-dimensional case, the operation of this filter can be described as follows. A one-dimensional slice of the input function is extracted from the three-dimensional input, along the line described by \( x = x_0 + v_xt \) and \( y = y_0 + v_yt \). Then the corresponding slice of the output function is formed by convolving the input slice with the one-dimensional function \( f(t) \). This is depicted in Figure 2.2.9.

This has extremely important implications in terms of the implementation of such a filter. Normally, a three-dimensional convolution would be required to implement the filter, but here, as a mathematical consequence of the constant-velocity assumption, the velocity-tuned filter can be implemented with a set of one-dimensional convolutions.

### 2.2.8 Effects of the Three-dimensional Filter

Just as in the two-dimensional case, the Fourier transform of the output of this filter is given by the product of the Fourier transforms of the input and of the impulse response of the filter.
Figure 2.2.9 Operation of 3D Velocity-Tuned Filter
Thus, in the case where the input function is a constant-velocity function, the Fourier transform of the output is as follows. Using the notation that $v_x^i$ is the $x$ component of the input velocity, etc., we get:

$$F(\omega_x, \omega_y) \delta_p \left( \omega_x v_x^i + \omega_y v_y^i + \omega_t \right) H \left( \omega_x v_x^f + \omega_y v_y^f + \omega_t \right)$$

Since the impulse plane is zero when its argument is non-zero, we can simplify the argument of $H$, by setting $\omega_x v_x^i + \omega_y v_y^i + \omega_t = 0$. Then the Fourier transform of the output becomes:

$$Y(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) \delta_p \left( \omega_x (v_x^f - v_x^i) + \omega_y (v_y^f - v_y^i) \right) + \omega_t \right)$$

$$Z(\omega_x, \omega_y) \delta_p \left( \omega_x v_x^i + \omega_y v_y^i + \omega_t \right)$$

where:

$$Z(\omega_x, \omega_y) = F(\omega_x, \omega_y) \delta_p \left( \omega_x (v_x^f - v_x^i) + \omega_y (v_y^f - v_y^i) \right)$$

Now, from Theorem 2.2.3, we conclude that $Y(\omega_x, \omega_y, \omega_t)$ must be a three-dimensional constant-velocity function, with the same velocity as the input. Specifically:

$$y(x, y, t) = z(x - v_x t, y - v_y t)$$

where $z(x, y)$ is the two-dimensional inverse Fourier transform of $Z(\omega_x, \omega_y)$. We can find an expression for $z(x, y)$ by applying the frequency multiplication property of the two-dimensional Fourier transform, as before, which gives:

$$z(x, t) = y(x, y) \ast \hat{h}(x, t)$$

where $\hat{h}(x, t)$ is the inverse Fourier transform of:

$$\hat{H}(\omega_x, \omega_y) \overset{\text{def}}{=} H \left( \omega_x (v_x^f - v_x^i) + \omega_y (v_y^f - v_y^i) \right)$$

62
We note that this has the form of a two-dimensional velocity-tuned filter, in \((\omega_x, \omega_y)\) space, rather than \((\omega_z, \omega_t)\). We can thus use Theorem 2.2.1 to derive the inverse Fourier transform. Depending on one's perspective, \(\hat{H}\) has the form either of a frequency-scaled version of \(H(\omega_x)\), translated in the \(\omega_y\) dimension, or of a frequency-scaled version of \(H(\omega_y)\), translated in the \(\omega_x\) dimension. Thus, there are two equivalent results:

\[
\hat{h}(x, y) = \frac{1}{|v^f_x - v^i_x|} h \left( \frac{x}{v^f_x - v^i_x} \right) \delta_l \left( y - x \left( \frac{v^f_y - v^i_y}{v^f_x - v^i_x} \right) \right) = \frac{1}{|v^f_y - v^i_y|} h \left( \frac{y}{v^f_y - v^i_y} \right) \delta_l \left( x - y \left( \frac{v^f_x - v^i_x}{v^f_y - v^i_y} \right) \right)
\]

Now, to see what these strange functions do in the convolution integral we refer to our previous discussion of two-dimensional velocity-tuned filters, and in particular to Theorem 2.2.2. Thus, in the same manner as the two-dimensional velocity-tuned filter, this convolution performs the following operation. Each "constant-velocity slice" of the function \(f(x, y)\), parallel to the line \(x(v^f_y - v^i_y) = y(v^f_x - v^i_x)\), is convolved with the one-dimensional function:

\[
h \left( \frac{x}{v^f_x - v^i_x} \right) \text{ or } h \left( \frac{y}{v^f_y - v^i_y} \right)
\]

Of course, these two functions are equal along the constant-velocity line.

Thus, this is similar to the two-dimensional case described earlier. Here the output of the filter is a constant-velocity function of the same velocity as the input. The function which is translating at constant velocity, namely \(f(x, y)\), gets smoothed along lines parallel to the line \(x(v^f_y - v^i_y) = y(v^f_x - v^i_x)\). The smoothing is done by a space-scaled version of \(h(x)\), where \(h(t)\) is the inverse Fourier transform of the filter generator function for the filter. The amount of smoothing is a function both of the bandwidth of the function \(h(t)\), and also the difference between the velocities. As the velocity difference increases, the smoothing also increases.
In the special case where the velocity of the input equals the velocity of the filter, the smoothing filter becomes an impulse, so no smoothing actually occurs. Thus the input and output of the filter are identical, within a scalar. This is confirmed by the frequency domain description of this case; with input and filter velocities equal:

\[ F(\omega_x, \omega_y) H(0) \delta_p \left( \omega_x v_x^i + \omega_y v_y^i + \omega_z \right) \]

In other words, the output of the filter equals the input, scaled by the constant, \( H(0) \).

Finally, if the input is an arbitrary, non-oriented function, it may be passed unchanged by the velocity-tuned filter, if the bandwidth of the filter generator function is large enough to accommodate the support of the input. This will be explored more deeply momentarily. When this is not the case, the output of the filter will tend to become oriented by virtue of the fact that the support of its Fourier transform will be oriented along the constant-velocity line of the filter, in three-dimensional frequency space. In the space-time domain, the filter will tend to smear, or smooth the input function along lines parallel to the constant-velocity line of the filter, thus causing the output of the filter to tend to appear as a constant-velocity function with the same velocity as the filter.

### 2.2.9 Effective Velocity Bandwidth of the Three-dimensional Filter

As in the two-dimensional case, it is relevant to discuss the bandwidth of the three dimensional filter. We assume that the one-dimensional filter generator function, \( F(\omega_z) \), has a temporal bandwidth \( B_z \).

We start with the case where the input is a constant-velocity function. We assume that the support of its Fourier transform is bandlimited to \( f_z < W_z \) and \( f_y < W_y \). To derive the effective velocity
bandwidth of the filter, consider Figure 2.2.10, where we have superposed the support of the input function with that of the filter. Again, we have delimited the support of the input and filter by their respective bandwidths.

With reference to Figure 2.2.10, we can write an equation to describe the planar support of the input spectrum, and also that of the upper and lower limits of the filter support, which are also planar.

\[
\begin{align*}
    f_x v_x^f + f_y v_y^f + f_t^i &= 0 \\
    f_x v_x^f + f_y v_y^f + f_t^u &= B_t \\
    f_x v_x^f + f_y v_y^f + f_t^l &= -B_t
\end{align*}
\]

Again, from Figure 2.2.10, we see that, at the critical point, \textit{i.e.} where the support of the input touches the limits of the filter support:

\[
\begin{align*}
    f_t^i(W_x, W_y) &< f_t^u(W_x, W_y) \\
    f_t^i(W_x, W_y) &> f_t^l(W_x, W_y) \\
    f_t^i(-W_x, W_y) &< f_t^u(-W_x, W_y) \\
    f_t^i(-W_x, W_y) &> f_t^l(-W_x, W_y)
\end{align*}
\]

We solve the first set of equations for the \(f_t\) variables, and substitute into the second set of equations. These four equations can then be solved simultaneously to give expressions for the \(x\) and \(y\) components of the input velocity:

\[
\begin{align*}
    v_x^f - \frac{B_t}{W_x} &< v_x^i < v_x^f + \frac{B_t}{W_x} \\
    v_y^f - \frac{B_t}{W_y} &< v_y^i < v_y^f + \frac{B_t}{W_y}
\end{align*}
\]
Figure 2.2.10 Determining Velocity Bandwidth of 3D Filter
Section 2.2.10

Thus, using the terminology introduced in the two-dimensional case, we have shown that the three-dimensional filter has an effective velocity bandwidth for each of the \( x \) and \( y \) dimensions. As long as the \( x \) and \( y \) components of the input velocity fall within their respective effective velocity bandwidths of the \( x \) and \( y \) components of the filter velocity, then the input will be passed relatively unchanged by the filter. This result can be extended to \( n \)-dimensional space, and of course gives the same result for the two-dimensional case as derived earlier.

Finally, we consider the case where the input is an arbitrary function, bandlimited to \( f_x < W_x \), \( f_y < W_y \) and \( f_z < W_z \). With reference to Figure 2.2.11, we see that the filter will pass the input nearly unchanged if the support of the input lies within the support of the filter. The critical point occurs at the point \((W_x, W_y, W_z)\), where we must have the following condition:

\[
\begin{align*}
    f_x^* (W_x, \pm W_y) &> W_z \\
    f_y^* (-W_x, \pm W_y) &> W_z \\
\end{align*}
\]

from which we can derive the requirement:

\[
\begin{align*}
    B_t &> W_t \pm W_x v_x^f \\
    B_t &> W_t \pm W_y v_y^f \\
\end{align*}
\]

or equivalently:

\[
\begin{align*}
    B_t &> W_t + \left| W_x v_x^f \right| \\
    B_t &> W_t + \left| W_y v_y^f \right| \\
\end{align*}
\]

Finally, this requirement may be a bit more stringent than necessary in some special cases. For example, if the Fourier transform of the input had a spherical support, the critical point would be closer to the origin than in the general case of a brick-shaped support.
Figure 2.2.11 3D Bandwidth Requirements for Arbitrary Input

Only the first quadrant is shown
2.2.10 Resolution Tradeoffs in Three-dimensional Case

There will be the same tradeoff between velocity resolution and smearing of arbitrary inputs as in the two-dimensional case. In general, as the velocity bandwidth of the velocity-tuned filter gets smaller, \( i.e. \) as the filter becomes more finely tuned to velocity, the filter will pass a smaller class of functions "nearly unchanged", and will more severely smear out those which do not belong to that class. In the limit as the velocity bandwidth approaches zero, only constant-velocity functions of the same velocity as the filter will be passed unchanged by the filter; all other functions will be smeared out indefinitely, along constant-velocity lines parallel to the constant-velocity line of the filter, \( i.e. \) \((x - v_xt, y - v_yt)\).

2.3 Reconstruction Using Velocity-Tuned Filters

In this section, we will consider the use of velocity-tuned filters for interpolation of sampled functions. As explained in Appendix A, sampling an arbitrary function results in replication of its Fourier transform along the frequency axis corresponding to the sampled variable. Then, for exact reconstruction of an arbitrary sampled signal, it is necessary and sufficient that the Fourier transform of the original function be isolated from its spectral replicates. This is done by passing the sampled function through a low-pass filter which eliminates the spectral replicates from the Fourier transform of the sampled function, and passes the spectrum of the original function unchanged. In practice, there is no filter that completely eliminates the spectral replicates, nor one which can pass the original spectrum unchanged, however there are many that can produce approximate reconstructions from sampled functions.

In the discussions which follow, we will only consider sampling of functions in time. In general, the results could easily be extended to include spatial sampling as well; whether the original spectrum is replicated along the temporal frequency axis, or spatial frequency axis, or both, is not important here. What is important is the geometry of the original spectrum and how this affects the use of velocity-tuned reconstruction filters; this geometry is preserved in all cases.
2.3.1 Two-dimensional Case

In Figure 2.3.3, we depict the Fourier transform of an arbitrary, bandlimited function, which has been sampled in time. Superposed on this, we have shown the support of a constant-velocity function. Once again we have assumed that the support is limited by the bandwidth of the filter, noting that this assumption is only true when the filter generator function of the filter is a rectangular pulse.

From this figure, it is clear that we can use a velocity-tuned filter to get an exact reconstruction from the samples of any bandlimited function, as long as the bandwidth of the filter is sufficiently large, and as long as the sampling rate is high enough to get sufficient separation of the spectral replicates. As we shall see momentarily, the amount of separation needed is a function not only of the bandwidth of the original spectrum, but also the velocity of the filter as well.

Many of the properties of a velocity-tuned reconstruction filter can be inferred from previous discussions about the space-time domain behavior of the filter. As discussed in Section 2.2.2, the space-time domain effect of filtering with a velocity-tuned filter is to convolve each one-dimensional "constant-velocity slice" of the input with the inverse Fourier transform of the one-dimensional function from which the velocity-tuned filter was generated. As discussed in Section 2.2, the generator function for velocity-tuned filter is always a low-pass filter. Thus a velocity-tuned reconstruction filter works by low-pass filtering, or smoothing the sampled function along lines which are parallel to the constant-velocity line of the filter. Clearly, as a result of this type of smoothing, the more closely the original function approaches a constant-velocity function of the same velocity as the filter, the better the reconstruction will be. On the other hand, if the filter has a relatively large bandwidth, then the smoothing will be less severe, and this restriction will be relaxed. Intuition tells us though that less smoothing can only work if the interval between samples decreases. Thus, in examining the requirements for a correct reconstruction with a velocity-tuned filter, we expect to see constraints between the velocities of the filter, the velocity, if any, of the input, the bandwidth of the filter and the sampling frequency. We will verify this shortly.
As shown in Figure 2.3.1, another way to view the space-time domain effect of the velocity-tuned filter on sampled functions, is that it tries to "connect the dots" between sample points, by drawing constant-velocity line through each sample point. When there are a series of points, spaced closely enough along a constant-velocity line, this process results in a single line connecting all of them.

2.3.2 Reconstruction of Two-dimensional Constant-velocity Inputs

First, we consider the case where the original, unsampled function is a constant-velocity function of arbitrary velocity. In Section 2.2.2, we discussed the effect of the velocity-tuned filter on unsampled functions of this type. We found that the output of the filter was a constant-velocity function of the same velocity as the input, but which was convolved in the spatial dimension by a time-scaled version of the filter generator function for the velocity-tuned filter. When the bandwidth of the velocity-tuned filter is large enough, relative to the bandwidth of the spectrum of the input function, the filter will pass the original spectrum approximately unchanged. The results of that discussion are directly applicable and extendable to this case, where the constant-velocity function has been sampled. Thus, if the bandwidth of the velocity-tuned filter is large enough to pass the original spectrum unchanged, and small enough to suppress the replicates of the original spectrum from the Fourier transform of the sampled function, then the original constant-velocity function will be approximately reconstructed from its samples. If the spectral replicates have been filtered out, but the bandwidth of the filter is not "large enough", then the output of the filter will be a reconstructed, low-pass filtered version of the original function.

The Fourier transform of a sampled constant-velocity function is shown in Figure 2.3.2. This figure suggests the conditions under which approximate reconstruction of the original function should be possible. Clearly, this can only work if the original function is bandlimited. In addition, the degree to which reconstruction is possible is a function of (1) the bandwidth of the velocity-tuned filter, (2) the bandwidth of the input, and (3) the sampling rate. At this point, we wish to quantify the relationship between these parameters.

In our discussion of effective velocity bandwidth, in Section 2.2.3, we derived a condition on the velocity of a constant-velocity input that would allow its spectrum to be passed with little change by
Figure 2.3.1  Constant-Velocity Filter "Connects the Dots" between sample values
Figure 2.3.2 Reconstruction of Constant Velocity Function with Velocity Tuned Filter
a velocity-tuned filter. That result assumed that the spectrum of the filter was approximately flat over its bandwidth. We now use that relation, given by Equation 2.2.4, to solve for the bandwidth that a velocity-tuned filter must have to pass a constant-velocity function of a velocity which may be different from the velocity of the filter. In particular, we find a constraint on the quantity $B_t$, which is the temporal bandwidth of the one-dimensional function used to generate the filter, and which as we defined earlier, is the relative temporal bandwidth of the two-dimensional velocity-tuned filter. Rearranging the previously derived constraint, gives us an equivalent pair of constraints:

$$B_t > W (v_i - v_f) \quad \text{and} \quad B_t > W (v_f - v_i) \quad 2.3.1$$

which can be expressed concisely as:

$$B_t > W |v_i - v_f| \quad 2.3.2$$

As defined earlier, the quantity $W$ is the spatial bandwidth of the constant-velocity function, which also equals the bandwidth of the one-dimensional function from which that function was generated.

Now, as indicated Figure 2.3.2, there is a further constraint between $B_t$ and the sampling frequency, namely that $B_t$ must be small enough to suppress the spectral replicates in the Fourier transform of the sampled function. As shown in Appendix E, this constraint can be expressed as:

$$B_t < f_s - W |v_i - v_f| \quad 2.3.3$$

Because $B_t$ is defined to be positive, this result has an important implication. When the difference between $v_i$ and $v_f$ becomes too large, then we get the impossible result that $B_t$ must be negative. The interpretation of this contradiction is, of course, that when these velocities become disparate enough, then there is no velocity-tuned filter that can both eliminate the spectral replicates and pass the original spectrum unchanged. The $W$ term in the above equation indicates this point is reached for smaller velocity differences as the bandwidth of the original function increases. This is certainly consistent with intuition and with Figure 2.3.2.
2.3.3 Minimum Sampling Rate for Two-dimensional Case

As was discussed in Section 2.1.1, one of the implications of the constant-velocity assumption, is that constant-velocity functions can be sampled without aliasing, as long as the sampling frequency is greater than zero. However, as suggested by Figure 2.3.2, there are constraints between the velocities of the input and of the filter, and the minimum sampling frequency needed to permit approximate reconstruction. Indeed, as shown in Appendix E, we can use the two previous results to find this minimum sampling rate. The result is:

\[ f_s > 2W \max \left| v_i - v_f \right| \]  \hspace{1cm} 2.3.4

This result is reminiscent of the Nyquist rate for sampling and reconstruction of arbitrary functions. It concisely expresses a result we have alluded to before. Namely, when the velocity of the filter and input are the same, we can sample a constant-velocity function at any non-zero rate, and still reconstruct the original function. As the velocities of the filter and input become more disparate, both the required sampling rate, and the bandwidth of the velocity-tuned reconstruction filter must be increased.

From these results, we draw the following conclusions about reconstruction of sampled constant-velocity functions. In the case where the velocity of the original function equals the velocity of the filter, it is clear that we will always be able to eliminate the spectral replicates from the Fourier transform of the sampled function, as long as the sampling frequency is greater than zero; all we need do is let the bandwidth of the filter approach zero. In addition, as discussed in Section 2.2.2, the gain of the velocity-tuned filter will be constant over the support of the spectrum of the original function, so that the spectrum is guaranteed to be passed unchanged by the filter, except for a possible scalar multiplication. Thus we conclude that reconstruction of a sampled constant-velocity function will always be possible using a velocity-tuned filter of the same velocity as the original function. In the framework suggested above, the filter just connects the samples of the function with a constant-velocity line, which in this case has the right velocity.
When the input function is a constant velocity function of a different velocity than the filter, approximate reconstruction may still be possible. Of course, we can always eliminate the spectral replicates by using a filter of small enough bandwidth. However, geometric constraints imposed by the velocities of the filter and of the original function dictate that in many cases, making the bandwidth of the filter small enough to kill the spectral replicates will also kill the high-frequency ends of the spectrum of the original function. More precisely, as was discussed in Section 2.2.2, the output of the filter will be a constant-velocity function with the same velocity as the original function, but the one-dimensional function which is being translated will be low-pass filtered, or smoothed, by the one-dimensional generator function of the filter. The amount of smoothing depends both on the bandwidth of the filter generator function, and on the difference between the input and filter velocities. As this difference increases, the amount of smoothing increases; as the difference goes to zero, the smoothing also decreases, and there is no smoothing when the velocities are the same.

2.3.4 Reconstruction of Arbitrary Two-dimensional Functions

Now, we consider the case where the original function is an arbitrary bandlimited function. In Figure 2.3.3, we depict the Fourier transform that results when we sample such a function, along with the support of a velocity-tuned filter. In Section 2.2.4, we saw that the velocity-tuned filter would pass such a function unchanged, under the right conditions. Thus, in the same manner as above, we extend this result by finding the condition under which the filter will eliminate the spectral replicates from the Fourier transform of the sampled function.

In Section 2.2.3, we found that if the input function had spatial bandwidth $W_z$ and temporal bandwidth $W_t$, then a velocity-tuned filter would pass the original spectrum approximately unchanged, if the temporal bandwidth of it filter generator function, $B_t$, satisfied:

$$B_t > W_t + v_f W_z$$

2.3.5
Figure 2.3.3 Reconstruction of Arbitrary Bandlimited Function With Velocity-Tuned Filter
Section 2.3.5  

Reconstruction Using Velocity-Tuned Filters

Now, it is shown in Appendix E that the bandwidth of the filter generator function must obey the following constraint, if it is to suppress the spectral replicates:

\[ B_t < f_s - W_t - \vert v_f W_x \vert \]

In like manner as before, these two constraints on \( B_t \) can be used to find the minimum sampling rate that will allow a velocity-tuned filter of specified velocity to be used for reconstruction. The result is:

\[ f_s > 2W_t + 2v_f W_x \]  2.3.6

Now, \( 2W_t \) is the Nyquist rate for sampling this function. That is, the classic Sampling Theorem tells us that if we want to temporally sample this function in such a way that it can be reconstructed from its samples, then the minimum sampling rate is \( 2W_t \). To reconstruct the original function, one would then use a low-pass filter whose bandwidth is the same as the bandwidth of the original function. However, the result obtained for a velocity-tuned reconstruction filter states analytically what is apparent from Figure 2.3.3, namely that the geometry of the support of such a filter makes it necessary to sample the original function at a rate exceeding the Nyquist rate, if exact approximate reconstruction is to occur.

2.3.5 When Two-dimensional Reconstruction Fails

For all of these cases, there are situations where the original function cannot be reconstructed from its samples by a velocity-tuned filter. In these cases, we are faced with a choice. We can either decrease the bandwidth of the filter enough to eliminate the spectral replicates, or leave the bandwidth high enough to pass all of the original spectrum.

In the first case, the higher-frequency components of the original spectrum will be killed. As we have seen, if the original function was a constant-velocity function, the output will be a constant-velocity
function of the same velocity, but just low-pass filtered in the spatial dimension. If the original function was an arbitrary function, its spectrum will be "oriented" by this operation, which will tend to make the output appear to be moving with constant velocity.

In cases where spectral replicates, or pieces thereof, are included in the filtered output, they will tend to add higher-frequency components to the reconstructed function, which will make it look "not as smooth" as the original function. Even though this phenomenon can result from sampling at a rate lower than the minimum sampling rate derived above, this is not aliasing. Aliasing is an irrevocable loss of information about the original function, embodied in a merging of spectral replicates in the Fourier transform of the sampled function. In this case, the spectral replicates are still distinct, and in fact the original function could be exactly recovered from the output of the filter in this case, merely by passing it through the "correct" reconstruction filter for the sampled function. The analogous situation in classical sampling theory is the case where the sampling rate exceeds the Nyquist rate, but the reconstruction filter has too large a bandwidth to correctly recover the original spectrum.

2.3.6 Reconstruction With Three-dimensional Velocity-tuned Filters

Now, we consider the use of three-dimensional velocity-tuned filters for interpolation of sampled functions. Most of our conclusions will come by direct analogy to the two-dimensional case. As before, sampling a three-dimensional function results in replication of the frequency spectrum of that function along the frequency axis or axes corresponding to the sampled variables. Again, we will only consider the case of temporal sampling. The job of the reconstruction filter is to eliminate the spectral replicates in the Fourier transform of the sampled function, and to pass the original spectrum unchanged.

Following the same argument as in the two-dimensional case, it is clear that any arbitrary, bandlimited function can be reconstructed from its samples by a velocity-tuned filter, if the sampling frequency is high enough, and if the bandwidth of the filter is large enough. Such a situation is depicted in Figure 2.3.6.
In the space-time domain, the velocity-tuned filter operates by convolving each one-dimensional “constant-velocity slice” of the three-dimensional input with the inverse Fourier transform of the one-dimensional function used to generate the three-dimensional filter. As in the two-dimensional case, this filter generator function will be a low-pass filter, thus the velocity-tuned filter functions by smearing or smoothing the sample values along lines parallel to the constant-velocity line of the filter. Thus, we may think of the filter as “connecting the dots” between sample points in three-dimensional space, as shown in Figure 2.3.4.

Of course, as in the two-dimensional case, the appropriateness of this filter for reconstruction of a sampled function depends on how much the original function looks like a constant-velocity function of the same velocity as the filter. Again, we will find that there are constraints between the bandwidth of the filter, the velocities of the filter and input, and the sampling frequency.

2.3.7 Reconstruction of Three-dimensional Constant-velocity Inputs

The three-dimensional case is a straightforward extension of the two-dimensional results. We have shown in Section 2.2.8 the conditions under which a velocity-tuned filter will pass a constant-velocity function of arbitrary velocity, relatively unchanged. To use the filter for reconstruction, we must determine in addition the necessary conditions under which the velocity-tuned filter will remove the replicates of the original spectrum from the Fourier transform of the sampled function.

In Figure 2.3.5, we depict the Fourier transform of a three-dimensional constant-velocity function, sampled in time. In Section 2.2.8, we derived conditions on the velocity of a constant-velocity function under which it would be passed approximately unchanged by a velocity-tuned filter. We now invert those results to find the conditions on $B_t$, the relative temporal bandwidth of the filter, under which the same thing happens. As in the two-dimensional case, $B_t$ is the temporal bandwidth of the one-dimensional filter generator function for the three-dimensional velocity-tuned filter.
Figure 2.3.4 Constant-Velocity Filter "Connects the Dots" between sample values
support of first spectral replicate bandlimited to \((\pm Wx, \pm Wy)\)

**Figure 2.3.5**: Reconstruction of 3D Constant Velocity Function using Velocity Tuned Filter
The constraints on $B_t$ thus derived are:

\[
B_t > W_z \left( v_x^i - v_x^f \right) \quad \text{and} \quad B_t > W_z \left( v_z^i - v_z^f \right) \quad \text{and} \\
B_t > W_y \left( v_y^i - v_y^f \right) \quad \text{and} \quad B_t > W_y \left( v_y^f - v_y^i \right) \quad 2.3.7
\]

which can be expressed concisely as:

\[
B_t > W_z \left| v_x^i - v_x^f \right| \quad \text{and} \quad B_t > W_y \left| v_y^i - v_y^f \right| \quad 2.3.8
\]

Here, $W_x$ and $W_y$ are spatial bandwidths of the three-dimensional constant-velocity function, which also equal the bandwidths of the two-dimensional function from which it was generated.

Now, in order to perform reconstruction with the velocity-tuned filter, $B_t$ must be small enough to guarantee that the spectral replicates will be filtered out of the Fourier transform of the sampled function. As shown in Appendix E, the constraint can be expressed as:

\[
B_t < f_s - W_z \left| v_x^i - v_x^f \right| \quad \text{and} \\
B_t < f_s - W_y \left| v_y^i - v_y^f \right| \quad 2.3.9
\]

As in the two-dimensional case, we see the possibility that one or both of these conditions will produce the contradictory result that $B < 0$, meaning that, for the given bandwidths, $W_x$ and $W_y$, and with the given input and filter velocities, it is not possible to construct a velocity-tuned filter that will both eliminate the spectral replicates and pass the original spectrum unchanged.
2.3.8 Minimum Sampling Rate for Three-dimensional Case

As in the two-dimensional case, if we wish to sample a constant-velocity function and reconstruct it with a velocity-tuned filter of the same velocity, we will always be able to do so, as long as the sampling rate is non-zero. However, when the two velocities are different, the sampling rate can no longer be as small. This is because a larger filter bandwidth will be needed to pass the original spectrum unchanged, which in turn makes it necessary to use a larger sampling frequency, to ensure that the spectral replicates fall outside of the passband of the filter. We can explicitly state the relationship between these variables by combining the previous results. As shown in Appendix E, the minimum sampling rate thus derived is given by:

\[
\begin{align*}
    f_s & > 2W_z \left| v'_{z} - v_s \right| \quad \text{and} \\
    f_s & > 2W_y \left| v'_{y} - v_s \right| 
\end{align*}
\]

From these results, we draw the same conclusions about reconstructing three-dimensional constant-velocity functions with velocity-tuned filters, as we did in the two-dimensional case. Briefly, when the input has the same velocity as the filter, exact reconstruction is always possible, assuming we have at least one sample. As in the two-dimensional case, the velocity-tuned filter smoothes the sample points along lines parallel to the constant-velocity line of the filter, and is thus able to correctly reconstruct such a function from any non-zero number of its sample points. When the original function is a constant-velocity function of a different velocity than the filter, will be able to correctly reconstruct the function if its velocity lies within the effective velocity bandwidth of the filter, and if the sampling frequency is high enough that the spectral replicates are kept out of the passband of the filter. When the sampling frequency is not high enough, the reconstructed function will be a constant-velocity function of the correct velocity, but it will be low-pass filtered, or smoothed, in the spatial dimension by the velocity-tuned filter. More precisely, as described in Section 2.2.6, the two-dimensional function which was being translated in the original function will be smoothed along lines parallel to the line \(x(v'_{y} - v'_{z}) = y(v'_{z} - v'_{x})\). The one-dimensional smoothing filter will derived from the inverse
Fourier transform of the filter generator function; the degree of smoothing will depend not only on the bandwidth of the filter generator function, but also on the disparity between the velocities of the input and the filter.

2.3.9 Reconstruction of Arbitrary Three-dimensional Functions

As in the two-dimensional case, the three-dimensional velocity-tuned filter can be used to reconstruct arbitrary, bandlimited three-dimensional functions. We saw previously that the velocity-tuned filter can pass the spectrum of such a function, approximately unchanged, under certain conditions. As before, we extend this result by finding the conditions under which the filter also eliminated the spectral replicates from the Fourier transform of the sampled function.

First, from Section 2.2.8, we can solve for the minimum bandwidth needed to pass the original spectrum, approximately unchanged. With reference to Figure 2.3.6, the constraint is:

$$B_t > \hat{W}_x v_z^f + W_y v_y^f + W_t$$

Further, it is easy to show by geometry the maximum bandwidth allowable, if the spectral replicates are to be effectively suppressed. The result, from Appendix E, is:

$$B_t < f_s - W_t - |v_z^f W_x| \quad \text{and}$$

$$B_t < f_s - W_t - |v_y^f W_y|$$
Figure 2.3.6 Reconstruction of Arbitrary 3D Bandlimited Function Using Velocity Tuned Filter
Finally, as before, these two constraints on $B_t$ can be used to find the minimum sampling rate needed to use a given velocity-tuned filter to reconstruction an arbitrary function from its samples:

$$f_s > 2W_t + 2\left|v_x^f \ W_x\right| \quad \text{and} \quad f_s > 2W_t + 2\left|v_y^f \ W_y\right|$$

We recognize $2W_t$ as the Nyquist rate for this arbitrary function. Thus, as in the two-dimensional case, use of a velocity-tuned reconstruction filter requires that we sample faster than we would need to if we used a standard low-pass filter.

**2.3.10 When Three-dimensional Reconstruction Fails**

As in the two-dimensional case, there are situations where the original function cannot be reconstructed from its samples by a velocity-tuned filter. In these cases, we are faced with the same choice as before. We can either decrease the bandwidth of the filter enough to eliminate the spectral replicates, or leave the bandwidth high enough to pass all of the original spectrum. In the first case, the higher-frequency components of the original spectrum will be killed. As in the two-dimensional case, if the original function was a constant-velocity function, the output will be a spatially low-pass filtered version of the input. If the original function was an arbitrary function, its spectrum will become "oriented" along the constant-velocity line of the filter, which can make the output appear considerably different from the original function, in particular, by making it tend to move at the velocity for which the filter is tuned. In cases where pieces of the spectral replicates are included in the filtered output, they will add higher-frequency components to the reconstructed function, making it look "not as smooth" as the original function. As in the two-dimensional case, since the entire original spectrum is included in the Fourier transform of this function, the original function can be correctly reconstructed by passing the function through a filter which removes the remaining spectral replicates.
2.4 A Regularized Solution to Spatio-temporal Approximation

Recently, many important visual tasks have been cast in the form of ill-posed problems [43]. In the sense of Hadamard [44], a well-posed problem is one whose solution (1) exists, (2) is unique, and (3) depends continuously on the initial data. Ill-posed problems, then, are those which are not well-posed.

As an example of an ill-posed visual problem, consider that of edge-detection. Edge-detection may be loosely defined as finding changes in image intensity which signal important physical changes in the field of view. It is clearly a fundamental problem in low-level vision, and one which is surprisingly difficult, in view of the case with which the human visual system performs it. As detailed by Torre and Poggio [43], a very likely first step of edge-detection is a differentiation of the image intensity values to localize intensity changes. However, measurements of image intensity are inherently noisy, and numerical differentiation is extremely sensitive to noisy data. Thus, edge-detection is an ill-posed problem because it fails to meet the third criterion for a well-posed problem.

2.4.1 Regularization of Ill-posed Problems

The approach to "solving" an ill-posed problem is to transform it into a well-posed problem which can be solved; this process is termed regularization. Regularization techniques for various types of ill-posed problems have been developed over the past few decades [45-47]. In general, regularization imposes constraints on the problem, in an attempt to force its solution to be well-defined. The regularized solution may then be chosen from the space of acceptable solutions by finding the function which minimizes a variational principle. As an example, the "inverse problem", of finding $x$ from data $y$ such that $Ax = y$, is often an ill-posed problem. Regularization of this problem requires choosing a norm, $|| ||$ and also a stabilizing functional $P$, the choice of which are usually based on knowledge of physical constraints on the problem, as well as mathematical tractability.
Given data \( y \), there are three main methods for finding the regularized solution \( z \):

1. Among \( z \) that satisfy \( ||Pz|| \leq C \), where \( C \) is a constant, find \( z \) such that the quantity \( ||Ax - y|| \) is minimized.

2. Among \( z \) that satisfy \( ||Ax - y|| \leq C \), find \( z \) such that the quantity \( ||Pz|| \) is minimized.

3. Find \( z \) such that the quantity \( ||Ax - y||^2 + \lambda||Pz||^2 \) is minimized.

In words, the first method finds the function that best approximates the data, \( i.e. \) minimizes the "distance" between the approximation \( z \) and the data, \( y \), subject to the constraint that \( ||Pz|| \leq C \). Conversely, the second method finds the most "regular" solution, subject to the constraint that the approximation lies sufficiently close to the data. The third method embodies a compromise between "closeness" and "regularity", and the regularizing parameter \( \lambda \) controls the relative strength or priority of these two constraints.

Existing results in standard regularization theory provides guidance in choosing the space of acceptable solutions, in finding the optimum \( \lambda \), and in choosing the stabilizing functional, \( P \), so that uniqueness and convergence of the result is guaranteed [47,48]. A commonly used class of stabilizers are the Tikhonov stabilizing functionals, defined as:

\[
||Pz||^2 = \int \sum_{r=0}^{p} p_r(x) \left( \frac{d^r z}{dx^r} \right)^2 dx
\]

Minimization of this term is equivalent to imposing a "smoothness" constraint on the solution, since it effectively minimizes the norm of the first \( r \) derivatives of the solution \( z \). This is widely applicable to physical problems, since it is observed that, over some scale, physical quantities tend to vary smoothly, as opposed to discontinuously.
2.4.2 Regularization of the Edge-detection Problem

The application of regularization to the edge-detection problem was proposed by Torre and Poggio, and expanded upon by Poggio, Voorhees and Yuille [49,50]. Given discrete samples of the intensity function, \( y_i \), it is desired to find a continuous function \( f(x) \), whose sample points \( f(x_i) \) approximate the data, and which is differentiable. They employ a scheme of the third type, formulating the problem as finding the approximation \( f(x) \), which minimizes:

\[
\sum_i (y_i - f(x_i))^2 + \lambda \int_{-\infty}^{+\infty} (f''(x))^2 \, dx
\]

The physical justification for a stabilizing functional of this form is that the intensity function falling on the retina must be smooth because it is bandlimited by the optics of the eye. Mathematically, it guarantees that the derivatives of \( f(x) \) exist and are bounded, and thus \( f(x) \) can be differentiated safely in order to locate intensity changes.

This problem has been considered in other disciplines, notably by Reinsch, for the task of regularizing numerical differentiation [51], and by Schoenberg [52]. Both investigators presented the solution in the form of approximating cubic splines. As shown by Poggio et al., when the data are given on a grid with regular spacing, then in many cases the approximating spline can be generated by convolving the data points \( y_i \) with a cubic spline. We stress that the approximating spline function approximates, rather than interpolates the data. An interpolating spline is constrained to pass through the data points, whereas an approximating spline is only constrained to "go near" the data.

Poggio et al., [50], also derive a regularized solution, based on the same regularizing functional, but assuming that the input data is continuous. The relation between the data and the solution is described in frequency space by the following Fourier transform:

\[
R(\omega) \overset{\text{def}}{=} \frac{Y(\omega)}{F(\omega)} = \frac{1}{1 + \lambda \omega^4}
\]  \hspace{1cm} 2.4.1
and in the space-domain (for \( x > 0 \)) by:

\[
    r(x) = \frac{\mu}{2\sqrt{2}} \exp \left( -\frac{x\mu}{\sqrt{2}} \right) \left[ \cos \left( \frac{x\mu}{\sqrt{2}} \right) + \sin \left( \frac{x\mu}{\sqrt{2}} \right) \right]
\]

where: \( \lambda \mu^4 = 1 \)

It was shown numerically that this function is essentially identical to the filter derived for the discrete-data case. It was also shown that this function is similar to the Gaussian function, and gives a sound theoretical basis for smoothing of image data with Gaussian-like filters prior to differentiation in edge-detection algorithms, such as was proposed by Marr and Poggio [20], and Marr and Hildreth [4] and implied by Canny [53].

Standard regularization, or regularization-like approaches have been applied in similar fashion to many early visual tasks, including computation of surface representations [54], computation of optical flow [56], and computation of shape from shading [55].

### 2.4.3 Regularization of the Spatio-temporal Approximation Problem

We now consider the problem of reconstructing a time-varying image, from its temporal samples. Of course, we have discussed this at some length in previous sections, but in doing so, we have made some explicit assumptions. In particular, we have assumed that the input and output of this task are well-behaved functions. In making our frequency domain arguments, we have made the assumption that the Fourier transforms of the input and output exist. This assumption is not valid for the stochastic signals, i.e. for those signals which are probabilistic, rather than deterministic. Of course, image intensity as measured by our retinas is a good example of a stochastic signal.

This is not to say that the previous discussions were without merit. There is an extensive literature on sampling and reconstruction of stochastic signals, and it seems fair to say that most of the "ideal case" results can be proven to be true for stochastic signals in some probabilistic sense, such as in the
mean-square sense [6]. Also, discussions of this type place an upper limit on performance of real-world interpolation systems.

Our approach is to assert that this problem is ill-posed. This is because (1) the reconstruction problem does not have a unique solution, since an arbitrarily large set of functions can be described which go through a fixed number of data points, and (2) the reconstruction result can be greatly influenced by noise in the sample data.

Our response to the situation will be to regularize the problem. The first task is the choice of an appropriate stabilizing functional. Once again, we will make a constant-velocity assumption, but this time, the assumption will be embodied in the stabilizing functional. In particular, we assert that the output of the regularized filter should be smooth along lines of constant velocity. To accomplish this mathematically, we will minimize the norm of the directional derivative of the image, along constant-velocity lines. Thus, the simplest stabilizing functional we use is:

$$PI = \int \int_{-\infty}^{+\infty} [\nabla_{\vec{a}} I]^2 \, dx \, dt$$

where $\nabla_{\vec{a}} I$ is the directional derivative of the image, $I(x,t)$, in the direction of a $\vec{a}$ which lies in the direction of the constant-velocity line: $\vec{a} = vt \hat{x} + t \hat{t}$. As shown in Appendix F, using this stabilizer, and assuming the data are continuous, the regularized solution to this problem is to pass the data through a linear filter, whose Fourier transform is:

$$H(\omega_x, \omega_t) = \frac{1}{1 + \frac{\lambda}{(v^2 + 1)} (\omega_t + v \omega_x)^2}$$

2.4.3
When a higher degree of smoothing is desired, we use the second directional derivative, along the constant-velocity line, inside the stabilizing functional:

\[ PI = \iint_{-\infty}^{+\infty} \left[ \nabla_\theta \nabla_\theta I^2 \right] dx \, dt \]

As shown in Appendix F, the regularized solution has the form of a linear filter, as before, whose Fourier transform is:

\[ H(\omega_x, \omega_t) = \frac{1}{1 + \frac{\lambda}{(v^2 + 1)^2} (\omega_t + v \omega_x)^4} \tag{2.4.4} \]

Of particular interest is that these filters have the form of velocity-tuned filters, as described earlier. In particular, we can write the second result as:

\[ H(\omega_x, \omega_t) = F(\omega_t + v \omega_x) \]

where:

\[ F(\omega_t) = \frac{1}{1 + \lambda' (\omega_t)^4} \tag{2.4.5} \]

\[ \lambda' = \frac{\lambda}{(v^2 + 1)^2} \]

Thus, we can invoke the results of the previous sections to determine the space-time domain behavior of the filter, to characterize the bandwidth of the filter, etc. In particular, we infer from previous results that the space-time effect of this filter is the convolution of one-dimensional constant-velocity slices of the input image, with \( f(t) \), the inverse Fourier transform of \( F(\omega_t) \).
Section 2.4.4  

A Regularized Solution to Spatio-temporal Approximation

In addition, \( F(\omega_t) \) has exactly the same form as the regularized one-dimensional edge-detection filter of Poggio, Voorhees and Yuille, for continuous data. From this similarity, we can infer two things. First, the solution for \( f(t) \) for the spatio-temporal approximation task is the same as that in the edge-detection task, as given by Equation 2.4.2. Because of the similarity between this function and the Gaussian, as exposed in the edge-detection analysis, this result would be supportive of the concept of filtering a time-varying image along constant-velocity lines, using a Gaussian filter, as an implementation of spatio-temporal approximation. Secondly, we can infer that the regularized solution for \( f(t) \), assuming discrete data will be the same as in the edge-detection case, i.e. that the general solution is a cubic approximating spline, and that this result can be produced by convolution with another cubic spline when the data are presented on a regular grid. Because of the equivalence of the continuous-data and discrete-data results, we will discuss only the continuous-data solution subsequently. For the exact form of the cubic approximating spline, the reader is directed to Appendix I of Poggio et al.

2.4.4 Regularization of Three-dimensional Spatio-temporal Approximation

To extend this result to the three-dimensional case, we use the same approach as above, minimizing the directional derivatives of the image, along constant-velocity lines in three-dimensional space. Thus, we use a stabilizing functional of the same form as before:

\[
PI = \int_{-\infty}^{+\infty} \int \left[ \nabla_{\vec{a}} I \right]^2 dx dt
\]

where \( \nabla_{\vec{a}} I \) is now the three-dimensional directional derivative of the image, \( I(x, y, t) \), in the direction of \( \vec{a} \), which lies in the direction of the constant-velocity line:

\[
\vec{a} = v_x t \hat{x} + v_y t \hat{y} + t \hat{z}.
\]
Section 2.4.4

A Regularized Solution to Spatio-temporal Approximation

As shown in Appendix F, the regularized solution in the case of continuous data is to pass the data through yet another linear filter:

\[
H(\omega_z, \omega_t) = \frac{1}{1 + \frac{\lambda}{(v_z^2 + v_y^2 + 1)} (v_z \omega_z + v_y \omega_y + \omega_t)^2}
\]  \hspace{1cm} 2.4.6

Thus, as in the two-dimensional case, the result is a velocity-tuned filter, this time in three-dimensional space:

\[
H(\omega_z, \omega_t) = F(\omega_t + v_z \omega_z + v_y \omega_y)
\]  \hspace{1cm} 2.4.7

where:

\[
F(\omega_t) = \frac{1}{1 + \lambda' (\omega_t)^2}
\]

\[
\lambda' = \frac{\lambda}{(v_z^2 + v_y^2 + 1)}
\]

As shown previously, in the space-time domain, the effect of such a filter is to convolve one-dimensional constant-velocity slices of the image with the inverse Fourier transform of \(F(\omega_t)\). Furthermore since, as in the two-dimensional case, the filter generator function for the velocity-tuned filter has the same form as the regularized one-dimensional edge-detection filter, we can directly apply the edge-detection results as derived by Poggio et al. to the task of spatio-temporal approximation in three dimensions.

This result highlights an extremely important difference between the regularized edge-detection filter and the regularized spatio-temporal approximation filter. The edge-detection filter presented above applies only to the task of one-dimensional edge-detection; more complicated functionals and solutions are required in two or more dimensions (see Appendix 3 of Poggio et al.). By contrast, the spatio-temporal approximation filter applies to a two or three (or \(d\)) dimensional task; in all cases, the
problem is reduced to a one-dimensional convolution along constant-velocity lines in the \(d\)-dimensional space.

2.4.5 General Regularized Solution to Spatio-temporal Approximation

From the derivations presented in Appendix F, we can infer the results of using the \(n\)th directional derivative in the stabilizing functional; the above results can be extended, in two and three dimensions, respectively, to:

\[
H(\omega_x, \omega_t) = \frac{1}{1 + \frac{\lambda}{(v^2 + 1)^n} (\omega_t + v \omega_x)^{2n}}
\]

\[
H(\omega_x, \omega_y, \omega_t) = \frac{1}{1 + \frac{\lambda}{(v_x^2 + v_y^2 + 1)^n} (v_x \omega_x + v_y \omega_y + \omega_t)^{2n}}
\]

Thus as before, these are both velocity-tuned filters, and the filter generator function for both of these filters has the form:

\[
F(\omega_t) = \frac{1}{1 + \lambda'(\omega_t)^{2n}}
\]

As pointed out by Terzopoulos [57], the \(d\)-dimensional version of this filter corresponds to the class of \(d\)-dimensional multivariate generalized spline functionals of Duchon [58], and Meinguet [59]. These functionals give rise to spline approximation solutions, and have several properties which make them attractive to the application of solving visual problems. In this one-dimensional case, the functional reduces to:

\[
\int \left( \frac{\partial^n f(t)}{\partial t^n} \right)^2 dt
\]

96
The regularizing functional used by Poggio et al. is a special case of this; compare Equations 2.4.9 and 2.4.1. This is presented to support and extend the concept that by now should be clear, namely that by transforming the $d$-dimensional spatio-temporal approximation problem into a one-dimensional problem, the constant-velocity assumption, as stated above, allows us to avail ourselves of previously well-studied one-dimensional problems and results in regularization theory. In particular in this case, we are using one-dimensional spline approximation in two or three-dimensional space, to perform the spatio-temporal approximation task.

An explicit time-domain expression for this general result is derived in Appendix F. The result, $h(t)$, can be expressed as the sum of causal and anti-causal exponentially damped sinusoids:

$$h(t) = \sum_{k=1}^{n/2} 2P \exp \{ s_k^* |t| \} \left[ s_k^* \cos s_k^* |t| - s_k^* \sin s_k^* |t| \right]$$

$$n \text{ even}$$

$$h(t) = \sum_{k=1}^{(n-1)/2} 2P \exp \{ s_k^* |t| \} \left[ s_k^* \cos s_k^* |t| - s_k^* \sin s_k^* |t| \right]$$

$$- P \omega_c \exp \{ -\omega_c |t| \}$$

$$n \text{ odd}$$

where:

$$\omega_c = \left( \frac{1}{\lambda} \right)^{1/n}$$

$$s_k = \omega_c e^{j \pi \left[ \frac{1}{2} + \frac{2k - 1}{2n} \right]} \quad k = 1, 2, \ldots 2n$$

$$s_k^* = \text{Re} (s_k) \quad s_k^i = \text{Im} (s_k)$$

$$P \overset{\text{def}}{=} \left[ (-\omega_c)^2 2^n \prod_{k'=1}^{n-1} \left( 1 - \cos \left[ \frac{\pi}{n} k' \right] \right) \right]^{-1}$$

97
This extends the result given by Poggio et al. for the \( n = 2 \) case. From this result, we see that the space-time duration of the filter is ultimately dominated by the exponential decay corresponding to those poles, \( s_k \), whose real parts have the smallest magnitude, \( i.e. \) those which lie closest to the imaginary axis. Now, \( s_1 \) is always one such pole, and it can be seen that it gets closer to the imaginary axis (\( i.e. \) it approaches \( \omega_c \, \exp\{j\pi/2\} \)), as the order of the filter, \( n \), goes up. This expresses what we expect from intuition. Imposing a higher-order smoothness constraint pushes the filter order up, which causes \( f(t) \) to decay more slowly; since it is \( f(t) \) which is used to convolve the input along constant-velocity lines, this results in more constant-velocity smoothing.

The effect of filter order, and of the parameter lambda, on the filter generator function \( F(\omega_t) \), and on the corresponding spline function, \( f(t) \), are illustrated in Figures 2.4.1 through 2.4.4.

### 2.4.6 Relation Between \( \lambda \) and Velocity

Another major difference between the edge-detection result and our result is that the effect of the regularizing parameter, \( \lambda \), is modulated by the velocity-norm, which is \( (v_x^2 + v_y^2 + 1)^n \) in the three-dimensional case. This expresses analytically another intuitive concept. As the velocity of the filter increases, we must increase \( \lambda \) as well, in order to maintain the same degree of smoothness. This can be readily seen by considering this filter in the context of a reconstruction filter for sampled constant-velocity functions. As discussed in the previous section, the action of a velocity-tuned reconstruction filter is to "connect the dots" along constant-velocity lines of the sampled image. As the velocity of the input increases, and for a fixed temporal sampling rate, the distance between the sampled points in the original image increases. Thus, to connect these points and thereby reconstruct the original image, the function with which we convolve these points must become smoother. This convolution function is of course \( f(t) \), the inverse Fourier transform of the filter generator function as given above. From the analytic time-domain solution, we see this dependence between \( \lambda \) and velocity explicitly. Specifically, as the velocity increases, \( \lambda' \) decreases, which causes the poles, \( s_k \), to move closer to the imaginary axis. As discussed above, this causes the the exponential envelope of \( f(t) \) to decay more slowly, resulting in an increases constant-velocity smoothing, as expected.
Figure 2.4.1 Regularized Filter of Orders 1, 2 and 3
Figure 2.4.2 Frequency Spectra of Regularized Filters of Orders 1, 2 and 3
Figure 2.4.3 Second Order Regularized Filters with Increasing Lambda
Figure 2.4.4 Frequency Spectra of Second Order Filters with Increasing Lambda
2.4.7 Relation to Butterworth Filters

Finally, it is shown in Appendix F, that the time domain description of generalized solution, given in Equation 2.4.8, can be expressed as the auto-correlation of Butterworth filters of order $n$, with cutoff frequencies $\omega_c$. Now, the Butterworth filter is the maximally flat approximation to the causal ideal low-pass filter, and is used in many signal-processing applications for that reason; see [74], and [60], for example. By maximally-flat, we mean that the squared-magnitude of the Fourier transform is as flat as possible in the passband. For an $n$th order filter, this implies that the first $(2n - 1)$ derivatives of the squared-magnitude are zero at zero-frequency. This can be shown by examining the Taylor Series expansion of the squared-magnitude response, as described in [74]:

$$|F_{BW}(\omega)|^2 = 1 - \omega^{2n} + \omega^{4n} - \cdots$$

and evaluating successive derivatives with respect to $\omega$. Butterworth filters also have the property of being monotonic in the passband and stopband. These properties are a consequence of the pole location of the magnitude response of the filter; the poles are placed symmetrically with respect to the real axis in the frequency plane.

Now, the pole location of the magnitude response of our generalized filter shares the same symmetry as the Butterworth filter. The difference is that the Butterworth filter has poles located only in the right half-plane, while our filter has poles in both the right- and left-half-planes; these poles lie symmetrically with respect to the imaginary axis in the complex-frequency plane. As a consequence of having both left- and right-half-plane poles, the general filter must be non-causal, if it is to be stable. Thus we suggest that the general spatio-temporal approximation filter is the maximally-flat approximation to the ideal non-causal low-pass filter.

To support this claim, we note that the Fourier transform of the spatio-temporal approximation filter has the same form as the squared-magnitude of the Fourier transform of the Butterworth filter. Therefore, the squared-magnitude response of the spatio-temporal approximation filter can be found
by multiplying the Taylor series for the Butterworth filter by its complex conjugate; the first few terms of the resulting series are:

$$|F_{SA}(\omega)|^2 = 1 - 2\omega^{2n} + 3\omega^{4n} - \ldots$$

so the first \((2n - 1)\) derivatives are zero, as in the case of the Butterworth filter.

In summary, the velocity-tuned filter approach to spatio-temporal interpolation and a regularized, spatio-temporal approximation solution are mutually supportive. The former suggests using a velocity-tuned filter, whose one-dimensional filter generator function is a low-pass filter; the latter concurs, and asserts that the specific form of the filter generator function should be this auto-correlated Butterworth-filter approximation to the ideal low-pass filter. The unifying concept, and the one that makes these results emerge from the mathematics, is the constant-velocity assumption.

2.5 Relaxing the Constant-Velocity Assumption

In previous sections, we have developed a fair amount of theory concerning a rather particular case, namely that when the image under consideration represents constant-velocity translation of a rigid body over all time. In this section, we will explore the consequences of relaxation of these rather stringent conditions. In particular, we will first examine the case of rigid-body translation with arbitrary velocity, and then look at finite-time rigid-body translation.

2.5.1 Rigid-Body Translation with Arbitrary Velocity

Let us first consider the two-dimensional case. If a two-dimensional image represents rigid-body translation with arbitrary velocity, we may express this analytically by:

$$g(x, t) = f(x - x_T(t))$$
We will call $x_T(t)$ the *translation function* for $g(x,t)$, for obvious reasons. The instantaneous velocity of the function is of course just the derivative of $x_T(t)$. To make a comparison to the constant-velocity case, we rewrite this expression as:

$$g(x,t) = f(x - vt - x_m(t))$$  \hfill 2.5.1

We will call $x_m(t)$ the *modulation function*, for reasons that will become apparent in a moment. This is a completely general form, inasmuch as $v$ can be made zero if $x_T(t)$ has no linear component. Now, as shown in Appendix G, the Fourier transform of the function is:

$$G(\omega_z, \omega_k) = F(\omega_z) P(\omega_z, \omega_k)$$  \hfill 2.5.2

where:

$$P(\omega_z, \omega_k) = \mathcal{F}_t \{ p(t) \}$$

$$p(t) = \exp \{-j[\omega_z vt + \omega_z x_m(t)]\}$$

We note that $\omega_z$ is a free parameter in $p(t)$, so that the one-dimensional, temporal Fourier transform $\mathcal{F}_t \{ p(t) \}$, is denoted by the two-dimensional function $P(\omega_z, \omega_k)$ in the frequency domain. Now, in general, $P(\omega_z, \omega_k)$ does not have a closed-from solution. However, $p(t)$ has the form of an *angle-modulated* function, which will allow us to draw on results from frequency- and phase-modulation theory to characterize its Fourier transform; see for example [41-42]. In particular, $p(t)$ may be thought of as a complex phasor whose fundamental frequency ('carrier frequency') is $\omega_z v$, and whose instantaneous phase is given by the second term, $\omega_z x_m(t)$. Thus $x_m(t)$ modulates the phase of the phasor, motivating our use of the term 'modulation function'.
2.5.2 Constant-velocity Translation with Sinusoidal Modulation

We now consider the case where the modulation function is sinusoidal. This may seem to be as restrictive a condition as constant-velocity translation, and in fact it is. However, it allows us to draw results directly from single-tone frequency modulation theory, and forms the basis for considering more general inputs. So, now suppose that \( x_m(t) = A \sin \omega_m t \). Then:

\[
p(t) = \exp \{-j [\omega_z vt + A \omega_z \sin \omega_m t]\}
\]

Now, as shown in Appendix G, the temporal Fourier transform of \( p(t) \) is given by:

\[
P(\omega_z, \omega_t) = \mathcal{F}(p(t)) = \sum_{n=-\infty}^{+\infty} J_n(A \omega_z) \delta_1(\omega_t + \omega_z v + n \omega_m)
\]

where \( J_n(\beta) \) is the \( n \)th -order Bessel function of the first kind and argument \( \beta \), and \( \delta_1() \) represents the impulse line function that we have seen earlier. We recall that \( \delta_1() \) is zero everywhere, except along the line described by setting its argument to zero; at each point along this line, we find a one-dimensional impulse. Let us spend a moment to get an idea what \( P(\omega_z, \omega_t) \) looks like. From the expression, we see that the function consists of a set of impulse lines in the two-dimensional \( (\omega_z, \omega_t) \) plane. The impulse line that corresponds to the \( n = 0 \) term is the same one we saw in previously, in the case of pure constant-velocity translation : \( \delta_1(\omega_t + v \omega_z) \). However, now there are impulse lines which form sidebands; these have the form \( \delta_1(\omega_t + v \omega_z \pm n \omega_m) \), which run parallel to the central line at \( n = 0 \).

Because \( n \) runs from positive to negative infinity, it would seem that there are an infinite number of sidebands in \( P(\omega_z, \omega_t) \), so it has an infinite spatial and temporal bandwidth. However, as we shall see, there is a structure to \( P(\omega_z, \omega_t) \) that will make the overall spectrum of the moving function spatially and temporally bandlimited when the original function is spatially bandlimited.

To demonstrate this structure, we need to consider the properties of the Bessel functions which act as coefficients for the sidebands. For \( \beta = 0 \), \( J_0(\beta) = 1 \), and all the other Bessel functions are...
zero. As $\beta$ increases, $J_0(\beta)$ decreases, the other Bessel functions start to increase, but at differing rates: $J_1(\beta)$ increases the fastest, followed by $J_2(\beta)$, and so on. Each Bessel function eventually goes through a maximum, and proceeds to oscillate non-periodically around zero. Thus, for any value of $\beta$, some Bessel functions are at or near their peak amplitude, others are going through zero, and still others, the higher-order ones, have not yet reached an amplitude which is "significant" with respect to the rest of them.

This behavior is quantified as follows. We will say that a Bessel function $J_n(\beta)$ has become significant if $|J_n(\beta)| > \epsilon$. Then the first $M$ Bessel functions are significant if $|J_M(\beta)| > \epsilon$ and $|J_{M+1}(\beta)| < \epsilon$. The function $M(\beta)$ can be graphed empirically, (see [41], for example), but it has been noted that, for $\epsilon = 0.01$, the curve can be approximated by:

$$M(\beta) \approx \beta + \alpha$$

where: $1 < \alpha < 2$

There is one more thing we need to know, and that is that the magnitude of $J_n(-\beta)$ equals the magnitude of $J_n(+\beta)$; when $n$ is odd, there is a sign change. From these properties, and by substituting $\beta = A \omega_z$, we can infer the following about the spectrum of $P(\omega_z, \omega_z)$. For each line, there is a region, centered around the $\omega_z$-axis, where the magnitude of its Bessel function coefficient is insignificant compared to the other coefficients. The size of the gap is related to the order $n$ of the corresponding Bessel function. Within this gap, we will make the assumption that the magnitude of the spectrum is zero.

An equivalent statement is that, for any $\omega_z$, the number of significant sideband pairs is given by the same expression for $M$; this is the number of significant Bessel function coefficients at that $\omega_z$, and this implies that both the $n = k$ and the $n = -k$ sidebands are significant. For very small $\beta$, we might conclude that only the central line is significant; however, if we included only the central line, the spectrum would represent pure constant-velocity translation; the first pair of sidebands is certainly required to carry the sinusoidal modulation component, and to be on the conservative side, we will always include the first two sideband pairs.
Section 2.5.2

Relaxing the Constant-Velocity Assumption

So now we have the following picture for the spectrum of $P(\omega_z, \omega_t)$. It consists of a number of parallel impulse lines, as we mentioned; for $A\omega_z \ll 1$, there the central line, and two pairs of sidebands, giving five significant lines. When $A\omega_z \approx 1$, an additional pair of sidebands becomes significant, for a total of seven. As $A\omega_z$ continues to increase, the number of pairs added continues to increase, linearly with $\omega_z$. A typical spectrum is shown in Figure 2.5.1. As shown in this figure, if we draw a line through the points at which these sidebands become significant, we will get a straight-line envelope; for values of $\omega_z$ which lie between the envelope and the $\omega_t$-axis, the spectrum is approximately zero.

There are two parameters which affect the location of this envelope. Clearly, as $A$ increases, there will be more significant sidebands at any given $\omega_z$; this is reflected as a narrowing of the gap around the $\omega_t$-axis, and a consequent increase in the magnitude of the slope of the envelope lines. In addition, as the modulation frequency increases, the spacing between the sidebands increases. This does not affect the number of significant sidebands at any $\omega_z$, but it does have the effect of narrowing the gap around the $\omega_t$-axis, just as increasing $A$ does.

Now, we are ready to see the relationship between the structure of $P(\omega_z, \omega_t)$, and the spectrum of the moving function. From Equation 2.5.1, the Fourier transform is given by:

$$G(\omega_z, \omega_t) = F(\omega_z) \sum_{n=-\infty}^{+\infty} J_n(A\omega_z) \delta_t(\omega_t + \omega_zv + n\omega_m)$$  \hspace{1cm} 2.5.5

From this analysis, it is clear that the spectrum of $P(\omega_z, \omega_t)$ cannot be passed unchanged by a filter of finite bandwidth, because it expands forever as $\omega_z$ increases. However, the spectrum of $G(\omega_z, \omega_t)$ is found by multiplying $P(\omega_z, \omega_t)$ by $F(\omega_z)$, the spectrum of the one-dimensional function that is moving around. Thus, if $F(\omega_z)$ is bandlimited, which it will be in practice, then the support of $G(\omega_z, \omega_t)$ will also be spatially bandlimited. In addition, the structure of $P(\omega_z, \omega_t)$ guarantees that $G(\omega_z, \omega_t)$ will also be temporally bandlimited in this case, so that a filter of finite bandwidth can be used. In particular, the resulting spectrum can be passed quite nicely by a velocity-tuned filter.
Figure 2.5.1 Constant Velocity Translation with Sinusoidal Modulation
of appropriate bandwidth; this situation is shown in Figure 2.5.2. We see from the figure that as the amplitude of the modulation becomes larger, there will be a larger number of significant sidebands within any given spatial bandwidth $W$, so the bandwidth of the velocity-tuned filter needed to pass the function unchanged will be larger. Of course, any filter with large enough bandwidth will suffice in this situation; however, in Section 2.5.4, we will argue that of all the suitable filters, a velocity-tuned filter may be optimal.

Let us now put aside the mathematics for a moment, and consider this situation qualitatively. We saw earlier that velocity-tuned filters with finite bandwidth can pass constant-velocity functions with little or no change when two conditions are met; the function being translated has to be bandlimited, and the velocity of the function has to be within the effective velocity bandwidth of the filter. That result makes this new result very reasonable. With the translation function we have assumed, $x_T(t) = vt + A \sin \omega_m t$, the velocity of the function is $v + A \omega_m \cos \omega_m t$; the velocity of the function is thus limited to the range $v \pm A \omega_m$. With the velocity bounded, and when $F(\omega_z)$ is bandlimited, we might expect that there is a velocity-tuned filter whose velocity bandwidth is large enough to pass the function approximately unchanged; this new result confirms that suspicion. In addition, we see that as $A$ or $\omega_m$ increase, the velocity range of the function increases so that, for a fixed spatial bandwidth $W$, the effective velocity bandwidth of the filter needed also increases; this result is reflected in the effects of $A$ and $\omega_z$ on the envelope as described above.

Finally, we consider two special cases, which may help increase our confidence in the mathematical result. First, consider the case when $A = 0$, i.e. when there is no modulation. Using the fact that $J_n(0) = 0$ for all $n \neq 0$, and $J_0(0) = 1$, the spectrum of the moving function reduces to:

$$G(\omega_z, \omega_t) = F(\omega_z) \delta(\omega_t + \omega_z v)$$  \hspace{1cm} 2.5.6

which is the result we derived earlier for constant-velocity translation, and thus the result we would insist upon obtaining.
Figure 2.5.2 Sinusoidal Modulation Of Bandlimited Function
Secondly, we stated earlier that if the translating function does not have a linear component, we could just let \( v = 0 \) in our assumed form of the translating function. We now are in a position to see that this indeed causes no problems with our formulation; the result is the same except that the central lobe and all the sidebands run parallel to the \( \omega_z \)-axis in the \( (\omega_z, \omega_t) \) plane. All the other results regarding envelopes and velocity-tuned filters still apply.

## 2.5.3 Constant Velocity Translation With Two Modulating Sinusoids

With the preceding result in hand, it becomes possible to consider more arbitrary translation functions. In particular, the natural tendency for some would be to decompose an arbitrary function into its frequency components, using Fourier techniques, and use superposition to find the resulting overall spectrum. However, we must point out that phase-modulation is not a linear transformation, so we should not expect this approach to work. To verify this prediction, we now examine the case where the modulating function consists of a pair of sinusoids. To the communications engineer, this corresponds to the case of two-tone phase modulation. With reference to Equation 2.5.2, we can write down the modulation function and the associated \( p(t) \):

\[
x_m(t) = A_1 \sin \omega_1 t + A_2 \sin \omega_2 t
\]

\[
p(t) = \exp \left\{ -j \left[ \omega_z vt + A_1 \omega_z \sin \omega_1 t + A_2 \omega_z \sin \omega_2 t \right] \right\}
\]

A derivation parallel to that of the single-frequency case would show that:

\[
P(\omega_z, \omega_t) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n(A_1 \omega_z) J_m(A_2 \omega_z) \delta_1(\omega_t + \omega_z v + n\omega_1 + m\omega_2)
\]

\[2.5.7\]
Now, we collect these terms into four groups:

1. The central line $\omega_t + \omega_x v = 0$ ("carrier component"), when $n = m = 0$.
2. A set of side-lobes at $\omega_t + \omega_x v + n\omega_1 = 0$, when $m = 0$.
3. A set of side-lobes at $\omega_t + \omega_x v + m\omega_2 = 0$, when $n = 0$.
4. A set of cross-terms at $\omega_t + \omega_x v + n\omega_1 + m\omega_2 = 0$, which includes all the other terms.

In other words, there are two sets of sidebands around the central lobe. The first set corresponds to $m = 0$, and the spacing between the lines is $\omega_1$; the second set corresponds to $n = 0$ and the spacing between those lines is $\omega_2$. In addition, there are cross-terms, which produce "sidebands around the sidebands". Thus, around each of the " $\omega_1$ " sidebands, there are sidebands spaced by $\omega_2$; of course, due to the symmetry in $n$ and $m$, an equivalent viewpoint is that there are sidebands centered around each of the " $\omega_2$ " sidebands, which are evenly spaced by $\omega_1$.

The support of such a $P(\omega_x, \omega_t)$ is shown in Figure 2.5.3. In the figure shown, we have assumed that the two frequencies are quite dissimilar, with $\omega_1 >> \omega_2$. This makes the structure of the sidebands more apparent by producing "clusters" around the $\omega_1$ sidebands; of course, these clusters extend indefinitely in either direction around the $\omega_1$ sidebands, so that the clusters merge and produce a fairly complex spectrum.

There are two important points to be made. The first is that that superposition does not hold in this situation. If superposition held, then $P(\omega_x, \omega_t)$ would be the sum of the spectra due to each of the sinusoidal terms. Then only the first three groups would be included only, i.e. a central line, and a set of lines contributed by each of the sinusoids. This is what we would see in an "amplitude modulation" scenario; however, phase-modulation is a non-linear process, so we get the cross-modulation terms expressed by the fourth set of terms.

However, there is an important conclusion that can be made without finding a closed-form expression for $P(\omega_x, \omega_t)$, and that is that if $F(\omega_x)$ is spatially bandlimited, then $G(\omega_x, \omega_t)$ will always be both spatially and temporally bandlimited.
Figure 2.5.3 Two-Tone Modulation -- Significant Sidebands

To see that this is true, we consider the two-tone case once more, and rewrite Equation 2.5.5 as:

\[
P(\omega_z, \omega_t) = \sum_{n=-\infty}^{+\infty} J_n(A_1 \omega_z) \left\{ \sum_{m=-\infty}^{+\infty} J_m(A_2 \omega_z) \delta_l(\omega_t + \omega_z v + n\omega_1 + m\omega_2) \right\}
\]

The intent is to express this as a set of sideband clusters, each one centered around \(\delta_l(\omega_t + \omega_z v + n\omega_1)\), and multiplied by the term \(J_n(A_1 \omega_z)\). We then assert that each sideband cluster is "significant" only if its coefficient, \(J_n(A_1 \omega_z)\) is significant, in the sense discussed above. Making use of the approximation presented above, i.e. that the number of significant Bessel functions \(J_n(A_1 \omega_z)\) is \(M \approx A_1 \omega_z + \alpha\), we conclude that the number of significant sideband clusters behaves in the same manner.

Now, at the value of \(\omega_z\) for which a particular sideband cluster becomes "significant", we will assume that all of the sidebands contained therein are significant if their coefficients \(J_m(A_2 \omega_z)\) are significant. We expect this is a conservative approximation. All these Bessel functions have value less than or equal to unity; thus the product of Bessel functions \(J_n(A_1 \omega_z) J_m(A_2 \omega_z)\) would be expected to make some sidebands in the clusters insignificant with respect to the sidebands around the central line. Nevertheless, the result of this discussion will not be affected by making this conservative assertion, so we will stick to it.

The main point of the discussion is that for any \(\omega_z\), there will always be a gap around the \(\omega_t\)-axis where there are no significant sidebands. This will cause \(G(\omega_z, \omega_t)\) to be spatially and temporally bandlimited when \(F(\omega_z)\) is bandlimited, as in the single-frequency case. To see that this assertion is true, we reason as follows. At any \(\omega_z = W_z\), there will be a finite number of significant sidebands of the form \(\delta_l(\omega_z + v \omega_z + n\omega_1)\), i.e. arising from the \(\sin \omega_1 t\) modulation term. Around each of these sidebands, there will be a significant sideband cluster; sideband clusters which are centered around insignificant sidebands are themselves considered to be insignificant. Now, each cluster, at the same \(\omega_z\), will in turn consist of a finite number of significant sidebands, so that the overall temporal bandwidth, \(W_t\), at that \(\omega_z\) will also be finite. Thus, if \(F(\omega_z)\) is spatially bandlimited to \(W_z\), then \(G(\omega_z, \omega_t)\) will also be temporally bandlimited to \(W_t\).
The temporal bandwidth of the overall function equals the sum of the bandwidth of the significant sidebands due to the $\sin \omega_1 t$ term is just the number of significant sidebands located at $\delta_t (\omega_x + v \omega_z + n \omega_1)$, plus the bandwidth added by the significant sidebands in the last significant clusters, i.e., the one for which $n$ is largest. Using the conservative approximation that the number of significant sideband pairs is $M \approx A \omega_z + 2$ as before, a good, though conservative estimate of the temporal bandwidth would be:

$$W_t = 2(A_1 \omega_z + 2)\omega_1 + 2(A_2 \omega_x + 2)\omega_2$$

The implication is that $F(\omega_x)$, and thus $G(\omega_x, \omega_t)$, are spatially bandlimited to $W_x$, then $G(\omega_x, \omega_t)$ is also temporally bandlimited, to $W_t(W_x)$. Thus the function can be passed nearly unchanged by a filter of the same spatial and temporal bandwidth. In addition, from $W_x$ and $W_t$ we could also determine the effective velocity bandwidth that a velocity-tuned filter would need to have, in order to pass this function nearly unchanged. The potential advantages of a velocity-tuned filter over other filters will be discussed in a moment.

But that is not what we mean to emphasize. The important point here is that even though superposition does not hold, the qualitative result of the single-frequency case still holds, namely that if $F(\omega_x)$ is bandlimited, then a velocity-tuned filter can be found which will pass the function $g(x, t)$ approximately unchanged. We stress that this is in intuitive agreement with the first result. It is true that adding another frequency component makes the velocity of the translating function more complex. However, it is still bounded to the range $v \pm (A_1 \omega_1 + A_2 \omega_2)$. Thus, as long as $f(x)$ is bandlimited, and the velocity with which it translates remains within the effective velocity bandwidth of a specified velocity-tuned filter, it should not be affected by the filter.

### 2.5.4 Constant-velocity Translation with Arbitrary Modulation

Finally, we briefly consider translation functions of arbitrary form. We will take the approach suggested earlier, considering an arbitrary function as the sum of its frequency components. Clearly, the
arguments of the previous section can be extended as the number of frequency components grows. To be sure, the situation becomes very complex; adding a third frequency component will add sidebands around the central line, around the central line of each sideband cluster, and around each of the sidebands within each cluster, and it gets worse as the number of sinusoids keeps increasing. However, as long as there are a finite number of sinusoids present, we can still use the same arguments as before; at a given \( \omega_z = W \), there will be a finite number of significant sidebands, sidebands of sidebands, sidebands of sidebands of sidebands, etc., so that the temporal bandwidth of \( P(\omega_z, \omega_t) \) at that \( \omega_z \) will also be finite.

Unfortunately, this argument breaks down for an arbitrary translation function, which is likely to have an infinite number of frequency components. Although each frequency component will contribute a finite number of significant sidebands at a given \( \omega_z \), there will be an infinite number of sidebands, giving rise to an infinite temporal bandwidth at each spatial frequency.

Well, this is not much to get upset about. First of all, it is reasonable that in the arbitrary case we should get to a point where finite-bandwidth filters are going to degrade the function in some way. After all, in the general case, we are now allowing things like step-changes in the translation function, which would require locally infinite velocities. Secondly, we can argue that many of the sidebands of sidebands of sidebands, etc., are likely to be insignificant since their coefficients are the products of several Bessel functions, the same argument we declined to make when the number of frequency components was finite.

Accepting the fact that we will have to lose some of the information in the function if we filter it, we argue that using a velocity-tuned filter is probably the optimal strategy. To see this, consider passing any \( G(\omega_z, \omega_t) \) through a velocity-tuned filter which is tuned to the same velocity \( v \) as used in the translation function. Clearly, the sidebands of \( G(\omega_z, \omega_t) \) which lie within the support of the velocity-tuned filter will be passed in their entirety. Indeed, the gain which multiplies each sideband will be constant over the support of the sideband. Furthermore, assuming that the filter generator function of the filter is symmetric, it will preserve the symmetry in the sidebands which arises from the symmetry of the Bessel functions, embodied in the property \( |J_n(\beta)| = |J_{-n}(\beta)| \). These properties
arise because the sidebands of the spectrum are aligned with the "axis" of the velocity-tuned filter, and would be lost otherwise. We do not offer a proof here, but suggest heuristically that these properties would result in a minimum amount of distortion of the function which is being translated, relative to other filters of comparable bandwidth.

In addition, we will make a prediction based on previous results. Namely, although we are allowing an arbitrary translation function, if the velocity which arises from such a function is bounded, and if the function \( f(x) \) which is being translated is bandlimited, then we expect that it can be passed with little or no change by a velocity-tuned filter of sufficient effective velocity bandwidth. If the velocity is bounded in this way, we infer that there is a limit to how rapidly the instantaneous displacement of \( f(x) \) can change; in other words, the translation function is bandlimited in this case. This further implies that there is may be a relationship between the bandwidth of the translation function, the bandwidth of the function \( f(x) \), and the effective velocity bandwidth of the velocity-tuned filter which would be needed to pass the function unchanged. Such a relationship would be similar to Carson’s Rule [41], an approximation which relates the bandwidth of a frequency-modulated signal to the bandwidth of the modulating function, and maximum frequency deviation which it produces.

At this point, it may be worthwhile to consider what has been said, in the context of the space-time domain. Recall that the effect of a velocity-tuned filter in space-time is a one-dimensional convolution of the image with a smoothing operator, along lines of constant velocity. As the bandwidth of the velocity-tuned filter gets smaller, \textit{i.e.} as the filter gets more sharply tuned to a particular velocity, the operator gets smoother, tending to smear things out over longer distances and times. Now, the previous result can be restated as follows. If the function \( f(x) \) is sufficiently bandlimited, \textit{i.e.} if it is "smooth enough", and if it is translated with a velocity which is close enough to the velocity of the velocity-tuned filter, then the smoothing effect of the filter will not be noticeable. As the range of velocities increases, the smoothing may become more noticeable, unless the bandwidth of the filter is increased, \textit{i.e.} unless the smoothing operator is made shorter and less smooth. This seems intuitively correct; hopefully our long journey through the frequency domain has not obscured this fundamental concept.
2.5.5 Three-dimensional Case

Let us now consider the three-dimensional case. One may expect the results to be essentially the same as the two-dimensional case, but it would seem prudent to verify this. If a three-dimensional image represents rigid-body translation with arbitrary velocity, we may express this analytically by:

\[ g(x, y, t) = f\left(x - x_T(t), y - y_T(t)\right) \]

where we now have two translation functions, \(x_T(t)\) and \(y_T(t)\). The instantaneous velocity of the function has two components, which are the derivatives of of the translation functions. Again, we rewrite this expression by pulling out the linear terms thus:

\[ g(x, y, t) = f\left(x - v_xt - x_m(t), y - vytt - y_m(t)\right) \]

where we now identify two modulation functions, one each for the \(x\) and \(y\) directions. Now, as shown in Appendix G, the Fourier transform of the function is:

\[ G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) P(\omega_x, \omega_y, \omega_t) \]

where:

\[ P(\omega_x, \omega_y, \omega_t) = \mathcal{F}\{p(t)\} \]

\[ p(t) = \exp\{-j[\omega_x v_xt + \omega_x x_m(t)]\} \exp\{-j[\omega_y v_yt + \omega_y y_m(t)]\} \]

In the same manner as the two-dimensional case, \(\omega_x\) and \(\omega_y\) are free parameters in \(p(t)\), so that the one-dimensional, temporal Fourier transform \(\mathcal{F}\{p(t)\}\), is denoted by the three-dimensional function \(P(\omega_x, \omega_y, \omega_t)\).
As before, we first consider the case where each of the modulation functions are sinusoids. Thus we have:

\[ x_m(t) = A_1 \sin \omega_1 t \]
\[ y_m(t) = A_2 \sin \omega_2 t \]
\[ p(t) = \exp \{ -j [\omega_x v_x t + A_1 \omega_x \sin \omega_1 t] \} \exp \{ -j [\omega_y v_y t + A_2 \omega_y \sin \omega_2 t] \} \]

Now, as shown in Appendix G, the temporal Fourier transform of \( p(t) \) is given by:

\[
P(\omega_x, \omega_y, \omega_t) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n (A_1 \omega_x) J_m (A_2 \omega_y) \times \delta_1 (\omega_t + \omega_x v_x + n\omega_1 + \omega_y v_y + m\omega_2)
\]

We see that the \( n = m = 0 \) term is \( \delta_1 (\omega_t + \omega_x v_x + \omega_y v_y) \). We recall that this is an impulse plane in \( (\omega_x, \omega_y, \omega_t) \) space; its support is a plane which goes through the origin, and whose orientation is governed by the two velocity components, \( v_x \) and \( v_y \). Thus, \( P(\omega_x, \omega_y, \omega_t) \) apparently consists of this "central plane", and of a set of "sidebands", which are also impulse planes, and which are parallel to the central plane. The sidebands which arise from the \( y \)-direction modulation are spaced by the \( y \)-direction frequency, \( \omega_1 \); those due to the \( x \)-direction modulation are spaced by \( \omega_2 \).

Again, because of the appearance of the Bessel functions, it is reasonable to assume that at any \( (\omega_x, \omega_y) \), there will be a finite number of sidebands which are significant. For the region where \( A_1 \omega_x << 1 \) and \( A_2 \omega_y << 1 \), there will be a gap around the \( \omega_t \)-axis where the only significant sidebands will be the first one or two around the central plane. As we move in the \( \omega_x \) direction, significant sidebands will appear at rate determined by the amplitude of the \( x \)-modulation, \( A_1 \); as we move in the \( \omega_y \) direction the same thing happens, except that the rate at which significant sidebands appear depends on \( A_2 \), the amplitude of the \( y \)-modulation. By analogy to the two-dimensional case, there will be a two-dimensional envelope around the \( \omega_t \)-axis, which demarcates the limits of the "significant" temporal bandwidth of \( P(\omega_x, \omega_y, \omega_t) \), at a particular \( (\omega_x, \omega_y) \).
From Equation 2.5.7, the overall spectrum of the moving function is:

\[ G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n(A_1 \omega_x) J_m(A_2 \omega_y) \times \delta_t(\omega_t + \omega_x v_x + n\omega_1 + \omega_y v_y + m\omega_2) \]

We note that for \( A_1 = A_2 = 0 \), this result reduces to the constant-velocity result we have seen before:

\[ G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) \delta_t(\omega_t + \omega_x v_x + \omega_y v_y) \]

The structure of \( P(\omega_x, \omega_y, \omega_t) \) is of course analogous to the one we saw in the two-dimensional case. Because of this, and because the overall spectrum is formed as the product of \( F(\omega_x, \omega_y) \) and \( P(\omega_x, \omega_y, \omega_t) \), we once again see that if the \( F(\omega_x, \omega_y) \), i.e. the spectrum of the function which is moving, is spatially bandlimited, then the spectrum that results from this sinusoidal modulation function will be both spatially and temporally bandlimited. By implication, such a function could then be passed through a filter of the appropriate bandwidth without significant change. In particular, a three-dimensional velocity-tuned filter as developed previously could be used if its effective velocity bandwidths, in both the \( \omega_x \) and \( \omega_y \) dimensions, were large enough.

It seems clear that the same arguments that were developed in the two-dimensional case for two-sinusoid modulation will also apply here. In particular, we know superposition will not hold, but we know there will always be a finite number of significant sidebands at a given \((\omega_x, \omega_y)\). Again, we need only argue that at any \((\omega_x, \omega_y)\), there will be a finite number of significant terms of the form \( \delta_t(\omega_t + \omega_x v_x + \omega_y v_y + n\omega_1) \). Around each of these sidebands, there will be a finite cluster of significant sidebands of the form \( \delta_t(\omega_t + \omega_x v_x + \omega_y v_y + n\omega_1 + m\omega_2) \). Thus, at any \((\omega_x, \omega_t)\), there will be a finite number of significant sidebands, giving rise to a finite temporal bandwidth.

In the case of arbitrary translation functions, the two-dimensional results will again apply. Thus, the spectrum of the moving object will in general have infinite bandwidth, so that no finite-bandwidth
filter will be able to pass the function unchanged. However, once again we will argue that of all possible bandlimited filters, the velocity-tuned filter seems to be optimal in some sense, for the same reasons cited earlier. Since the sidebands will be lined up along the axes of the velocity-tuned filter, the gain of the filter along each sideband is constant, and again, the symmetry of the sidebands will be preserved by the filter if the filter generator function is symmetric.

Finally, intuitively, there should be no qualitative difference between the two-dimensional and the three-dimensional cases. We found that three-dimensional velocity-tuned filters have effective velocity bandwidths in the $\omega_x$ and $\omega_y$ dimensions, and that if the function that is being translated is bandlimited in $\omega_x$ and $\omega_y$, it could be passed by a velocity-tuned filter approximately unchanged as long as the $x$ and $y$ components of the velocity were within those bandwidths. Thus, in the case of arbitrary motion, we can argue along the same lines; as long as $F(\omega_x, \omega_y)$ is bandlimited, and as long as the $x$ and $y$ components of the instantaneous velocity are bounded, i.e. the $x$ and $y$ translation functions are bandlimited, then there is a velocity-tuned filter which will pass the image of the translating function with little or no change.

2.5.6 Temporal Windowing

The results of the previous section would seem to be fairly general, but in fact there was an implicit assumption which made everything work out. This assumption was that the translation function $x_T(t)$ had a linear component $vt$ which had a constant velocity over all time. It is this assumption that made the spectrum oriented along the constant-velocity line $\omega_t + v \omega_x = 0$ in the frequency plane.

However, this assumption cannot be regarded as generally true. A more reasonable assumption would be that over some finite time, the linear component of an object's velocity is constant, and then it changes. In fact, it will often be more reasonable to bypass the formulation in the previous section, and claim that the whole velocity function behaves in this manner; in other words, we would say that the velocity is piece-wise constant. This is the essence of the constant-velocity assumption as formulated by Fahle and Poggio [36].
As usual, we want to see how this behavior is reflected in the frequency domain. Our approach is straightforward; we replace our constant-velocity function by the sum of many constant-velocity segments; these represent constant-velocity translation over a finite amount of time. Thus:

\[
g(x, t) = \sum_{n=-\infty}^{+\infty} g_n(x, t)
\]

\[
g_n(x, t) = \begin{cases} 
 f(x - v_n t) & t_n < t < t_{n+1} \\
 0 & \text{else}
\end{cases}
\]

Now, an equivalent way of expressing each of the constant-velocity segments \(g_n(x, t)\) is by writing them as the product of a constant-velocity function, which extends indefinitely, and a temporal window function, which is non-zero only over a finite time interval. Thus, we could rewrite the previous equation as:

\[
g_n(x, t) = f(x - v_n t) w(t) \quad 2.5.10
\]

\[
w(t) = \begin{cases} 
 1 & t_n < t < t_{n+1} \\
 0 & \text{else}
\end{cases}
\]

Now, superposition does hold in this case, so we can find the Fourier transform of \(g(x, t)\) by adding up all the Fourier transforms of the constant-velocity segments, \(g_n(x, t)\). As shown in Appendix G, these individual transforms are given by the expression:

\[
G_n(\omega_x, \omega_t) = F(\omega_x) W(\omega_t + v_n \omega_x) \quad 2.5.11
\]

We stress that this is a general result, not limited to the form of \(w(t)\) shown above. As an example, consider the case where \(w(t) = 1\) for all \(t\). Then we have:

\[
W(\omega_t) = \delta(\omega_t)
\]

which gives:

\[
G_n(\omega_x, \omega_t) = \delta_t(\omega_t + v_n \omega_x)
\]

123
Clearly, this is the case of pure constant-velocity translation for all time, so we should get the result we have seen before, and we do.

Now, in the case where \( w(t) \) has the form of a box function of duration \( \Delta t \), we can write:

\[
w(t) = \begin{cases} 
1 & t_n < t < t_n + \Delta t \\
0 & \text{else}
\end{cases}
\]

for which:

\[
W(\omega_t) = \Delta t \exp \left\{ -j \omega_t (t_n + \Delta t/2) \right\} \frac{\sin \omega_t \Delta t/2}{\omega_t \Delta t/2}
\]

\[
G_n(\omega_z, \omega_t) = F(\omega_z) \Delta t \exp \left\{ -j(\omega_t + v_n \omega_z)(t_n + \Delta t/2) \right\} \frac{\sin(\omega_t + v_n \omega_z) \Delta t/2}{(\omega_t + v_n \omega_z) \Delta t/2} \tag{2.5.12}
\]

The exponential term in the last equation is just a phase factor, which reflects the displacement of this constant-velocity segment from the origin of the \((x, t)\) plane. We are more concerned with the magnitude of \( G_n(\omega_z, \omega_t) \). In the case of constant-velocity translation for all time, \( G_n(\omega_z, \omega_t) \) was the product of \( F(\omega_z) \) and an impulse line, which as was mentioned, could be viewed as a one-dimensional impulse traversing the \((\omega_z, \omega_t)\) plane with constant velocity. In this case, \( G_n(\omega_z, \omega_t) \) is the product of \( F(\omega_z, \omega_t) \) and a “traveling sinc function”, moving at constant velocity in the \((\omega_z, \omega_t)\) plane.

If the window duration is infinite, then these two results should be the same, and they are; the sinc function behaves like an impulse function when \( \Delta t \) approaches infinity. In the general case, however, \( \Delta t \) is finite, and the sinc function “spreads out” horizontally from the constant-velocity line \((\omega_t + v_n \omega_z) = 0\). Now, the sinc function extends indefinitely, so if \( \Delta t \) is finite, then the spectrum \( G_n(\omega_z, \omega_t) \) covers the entire \((\omega_z, \omega_t)\) plane. However, the magnitude of this sinc function decreases as the inverse of the horizontal distance to the constant-velocity line \((\omega_t + v_n \omega_z) = 0\). Thus, there will be a symmetric region around this line where the “most significant part” of the spectrum is concentrated.

Clearly, there is no finite-bandwidth filter that can pass such a function without changing it. However, of all the finite-bandwidth filters, one would expect that a velocity-tuned filter would cause
the least distortion of the function, because of the geometry of the situation. To verify this prediction, consider a velocity-tuned filter which is tuned to the same velocity as the constant-velocity segment. We recall that the system function of a constant-velocity function has the form of a one-dimensional filter generator function \( H(\omega) \), translated in the frequency plane at constant velocity, thus: \( H(\omega_t + v_n \omega_x) \).

Now, if we pass a constant-velocity segment through such a filter, the Fourier transform of the constant-velocity segment gets multiplied by this system function. Thus the Fourier transform of the filter output would be:

\[
Y(\omega_x, \omega_t) = F(\omega_z) \cdot W(\omega_t + v_n \omega_x) \cdot H(\omega_t + v_n \omega_x)
\]

\[
= F(\omega_z) \cdot W_2(\omega_t + v_n \omega_x)
\]

where:

\[
W_2(\omega_t + v_n \omega_x) = W(\omega_t + v_n \omega_x) \cdot H(\omega_t + v_n \omega_x)
\]

Thus, the output of the filter is another constant-velocity segment; it has the same velocity as the input, which is not surprising. A more important point is demonstrated by taking the inverse Fourier transform of \( Y(\omega_x, \omega_t) \) to find:

\[
y(x, t) = f(x - v_n t) \cdot w_2(t)
\]

2.5.13

where:

\[
w_2(t) = w(t) \ast h(t)
\]

Here we have used the fact that the multiplication of \( W(\omega_t) \) and \( H(\omega_t) \) in the frequency domain is equivalent to the convolution of their respective inverse Fourier transforms. The important point is that the only difference between the output of the filter and the input is that a different temporal window has been used; it is the convolution of the original temporal window and the inverse Fourier transform of the filter generator function used to make the velocity-tuned filter. The function which is actually moving around, \( f(x) \), is not changed at all; there is absolutely no loss of spatial information in this function.
Of course, this result only appears in the case where the velocity of the filter exactly equals the velocity of the constant-velocity segment. However, on the basis of the results derived in Section 2.2, we can predict the result when these two velocities are not equal. In particular, if the significant sidelobes of the sinc function fall within the bandwidth of the velocity-tuned filter, then there should be little noticeable effect; the effect, when it appears, will be a smoothing of the function \( f(x) \), producing a graceful, rather than drastic degradation of the function.

### 2.5.7 Three-Dimensional Case

Briefly, we will show that these concepts can be extended to the three-dimensional case. We picture a three-dimensional moving function as being composed of many constant-velocity segments:

\[
g(x, y, t) = \sum_{n=-\infty}^{+\infty} g_n(x, y, t)
\]

\[
g_n(x, y, t) = \begin{cases} 
  f(x-v_{nx}t, y-v_{ny}t) & t_n < t < t_{n+1} \\
  0 & \text{else}
\end{cases}
\]

And again, this can be rewritten as the product of an infinite-duration constant-velocity function, and a finite-duration window function. In fact, it is the same window function as before:

\[
g_n(x, y, t) = f(x-v_{nx}t, y-v_{ny}t) w(t)
\]

\[
w(t) = \begin{cases} 
  1 & t_n < t < t_{n+1} \\
  0 & \text{else}
\end{cases}
\]

As shown in Appendix G, the Fourier transform of \( g_n(x, y, t) \) is given by the expression:

\[
G_n(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) W(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y)
\]
Now, we can extend the two-dimensional results to describe this result. When $w(t) = 1$, then $W(\omega_t)$ is a one-dimensional impulse, and the support of $G_n(\omega_x, \omega_y, \omega_t)$ is an impulse plane as we have seen earlier. When $w(t)$ has finite duration, $W(\omega_t)$ has infinite extent, so $G_n(\omega_x, \omega_y, \omega_t)$ spreads out around this plane, i.e. the one described by $\omega_t + v_{nx} \omega_x + v_{ny} \omega_y = 0$. In particular, when $w(t)$ has the form of a box function, $W(\omega_t)$ is a sinc function, so $G_n(\omega_x, \omega_y, \omega_t)$ is concentrated in the first several sidelobes around this plane. Thus, a velocity-tuned filter whose bandwidth is large enough to accommodate these sidelobes can, in principle, pass the "most significant part" of this spectrum unchanged.

To be more specific, we recall that a three-dimensional velocity-tuned filter is made by translating a one-dimensional filter generator function $H(\omega_t)$ through three-dimensional space: $H(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y)$. Thus, if we pass a constant-velocity segment through such a filter tuned to the same velocities $v_{nx}$ and $v_{ny}$, the Fourier transform of the filter output would be:

$$Y(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) W(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y) H(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y)$$

$$= F(\omega_x) W_2(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y)$$

where:

$$W_2(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y) = W(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y) H(\omega_t + v_{nx} \omega_x + v_{ny} \omega_y)$$

Thus, the output of the filter is another constant-velocity segment; it has the same velocity as the input. And the output of the filter is the product of the original constant-velocity function, and a new temporal window:

$$y(x, t) = f(x - v_{nx}t, y - v_{ny}t) w_2(t) \quad 2.5.16$$

where:

$$w_2(t) = w(t) \ast h(t)$$

As in the two-dimensional case, the only thing that is changed in this case is the window function. The function which is moving, $f(x)$ is not affected, so there is no loss of spatial resolution. When the
velocities of the filter and the constant-velocity segment are not equal, then we would expect smoothing of \( f(x) \), to an extent which depends on the relative bandwidths of the function and the filter, and on the difference in the velocities; we would not expect more drastic distortion of the function.

2.5.8 Predictions for Real-World Motion

In the formulations above, we pictured the velocity of the function as being piece-wise constant. However, that seems both unrealistic and unnecessarily pessimistic. The sharp changes in velocity that are implied in this formulation are what give rise to the sinc function, with its rather extensive sidelobes, in the frequency spectrum. However, from observation, one expects that while velocities in the real world do not tend to stay constant, they do tend to vary smoothly, with the exception of velocity discontinuities that occur, for example, when an object disappears behind an occluding object, or when a bouncing object suddenly changes direction.

A more exhaustive approach would take into account the smoothness of velocity; perhaps a combination of the phase-modulation and temporal windowing approaches would produce a neat description in the frequency domain. However, we can already intuit the result. In the three-dimensional case, the spectrum of the moving function will be oriented around the constant-velocity plane corresponding to its local average velocity; because the velocity is changing, the spectrum would not lie on a single plane in the frequency plane, but would be concentrated around that plane. Thus a velocity-tuned filter, tuned to a velocity near this local average velocity, and having a bandwidth approximately as large as the spectrum of the moving function, would pass it nearly unchanged. If the velocities were too disparate, or if the bandwidth of the filter was too small, then the function would be degraded "gracefully"; it would be smoothed by the filter, but not grossly distorted. Consequently, a velocity-tuned filter would seem to be a very good type of filter to use in this situation.
Detection of Moving Objects in the Presence of Noise: A Psychophysical Experiment

Mathematics is nice, but at this point, we are ready for a respite; we now shift our focus to implementation of the theory presented in Chapter 2, so that we can observe the properties of velocity-tuned filters on images, rather than on equations. We begin with a brief discussion of implementation methods themselves, and then proceed to results.

3.1 Implementation Issues

For the purposes of this thesis, implementation is synonymous with computer programming. This is because computers afford a relatively painless way to implement and evaluate ideas about vision, and also because the field of computer vision is an important application of vision research. Thus, our intent is to implement these ideas as computer-based image processing programs, whose output form, in a sense, experimental results. However, before we proceed with the implementation and results, there are a few issues that should be mentioned.

The most fundamental issue is that digital computers only deal with discrete signals, whereas all of our theoretical development has been in the continuous domain. Thus, we will need to develop discrete-domain, or digital filters which implement the continuous-domain, or analog filters which we have been discussing.
As in the continuous case, a digital filter can be implemented in either the space-time domain, or in the frequency domain. In the space-time domain, one constructs a discrete function which represents the impulse response of the desired filter; the filtering operation then consists of a discrete convolution of the signal with the filter impulse response. The alternative approach is to construct a discrete function which represents the system function of the desired filter. Then to filter a signal, one would compute its discrete Fourier transform, multiply the result by the system function of the filter, and take the inverse discrete Fourier transform of the product. Although this may seem somewhat less direct than the space-time approach, it is often faster, due to the existence of fast methods of computing the discrete forward and inverse Fourier transforms, as discussed in [60].

We have chosen to implement velocity-tuned filters in the space-time domain, for a number of reasons. One reason is that the space-time description of this filter, i.e. as an operator which smoothes an input image along constant-velocity lines, seems more intuitive than the frequency domain description, and we wish to emphasize this point. As we will discuss in Chapter 5, the space-time operator is well-suited implementation in a biological vision system. There are two other, more quantitative reasons, which we now present.

### 3.1.1 Computational Effort

Firstly, the computational advantage of the discrete frequency domain is lost in this particular case. To see why this is so, let us suppose that we wish to filter an discrete image \( I[m, n, t] \), which has dimensions \( M_t \) by \( N_t \) by \( T_t \), with a discrete filter \( h[m, n, t] \), which has dimensions \( M_f \) by \( N_f \) by \( T_f \). In general, the result of the filtering operation will have dimensions equal to the sum of the dimensions of the input and the filter, regardless of which approach is taken.

In the space-time domain, the filtering operation is given by the three-dimensional convolution:

\[
Y[m, n, t] = \sum_{t'=0}^{T_f-1} \sum_{m'=0}^{M_f-1} \sum_{n'=0}^{N_f-1} I[m', n', t'] h[m - m', n - n', t - t']
\]
Computing each point in the output requires multiplication by each point in the filter, so the total operation would require on the order of:

\[
\frac{(M_i + M_f)(N_i + N_f)(T_i + T_f)}{\text{points in output}} \times \frac{M_f N_f T_f}{\text{points in filter}}
\]

real multiplications.

Now, a frequency domain implementation would require that we compute the discrete Fourier transform of the input image, multiply it by the system function of a discrete velocity-tuned filter, and find the inverse discrete Fourier transform of the result. The three-dimensional discrete Fourier transform is defined by:

\[
I[k, l, p] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{t=0}^{T-1} I[m, n, t] \exp \left\{ -j2\pi \frac{ln}{N} \right\} \exp \left\{ -j2\pi \frac{km}{M} \right\} \exp \left\{ -j2\pi \frac{pt}{T} \right\}
\]

Thus, each point in the discrete Fourier transform of the input image requires \( M_i N_i T_i \) complex multiplications, and the whole computation requires:

\[
M_i N_i T_i \times M_f N_f T_f
\]

complex multiplications. However, to perform the filtering operation described above, we have to pad the image with zeros before computing the discrete Fourier transform; padding is required because multiplication of discrete Fourier transforms is equivalent to circular convolution in the space-time domain; in order to do linear convolution, which more closely resembles convolution in the continuous domain, the input image must be padded with zeros to the same dimensions as the linear convolution result. Thus, a frequency domain implementation that just cranks the discrete Fourier transform definition requires on the order of:

131
\[
\frac{(M_i + M_f)(N_i + N_f)(T_i + T_f)}{\text{points in output}} \times \frac{(M_i + M_f)(N_i + N_f)(T_i + T_f)}{\text{multiplications per point}}
\]

complex multiplications to compute the discrete Fourier transform. A fast transform technique such as
the FFT can reduce this number to :

\[
(M_i + M_f)(N_i + N_f)(T_i + T_f) \times \log_2(M_i + M_f)\log_2(N_i + N_f)\log_2(T_i + T_f)
\]

complex multiplications, however, the dimensions \((M_i + M_f), (N_i + N_f)\) and \(T_i + T_f\) must all
be powers of two; similar but smaller reductions may be possible in other cases. Performing the entire
filter operation would require on the order of twice this number of multiplications: one for the forward
transform, and one for the inverse transform.

Now, in the space-time domain, we saw that the velocity-tuned filter can be implemented by
performing one-dimensional convolutions along lines of constant velocity in a two-, three-, or \(n\)-
dimensional image. In particular, these constant-velocity slices are to be convolved with \(f(t)\), the
inverse Fourier transform of the filter generator function of the velocity-tuned filter. Thus, instead of
\(M_f N_f T_f\) multiplications per point, as we had before, now there are only \(T_f\) multiplications; in order
to perform this operation in the discrete domain, we need on the order of :

\[
\frac{(M_i + M_f)(N_i + N_f)(T_i + T_f)}{\text{points in output}} \times \frac{T_f}{\text{mult. per point}}
\]

real multiplications. In many applications, including ours, this number will be less than the number of
multiplications required to compute the three-dimensional discrete Fourier transform; only when the
images are very small spatially, and long temporally, could we lose badly. In addition, in the space-time
domain, one can stick pretty much to real, integer arithmetic, whereas using discrete Fourier transforms,
one clearly requires complex, floating-point arithmetic, further increasing the computational burden.
In principle, the geometry of the support of a velocity-tuned filter would make it unnecessary to compute the entire discrete Fourier transform of an image; however, it seems unlikely that one could gain significant reductions in the number of computations needed, beyond the number needed by the fast transform techniques. One could also suggest that the one-dimensional convolutions which embody the velocity-tuned filter could be sped up by doing them in the discrete frequency domain. However, in practice, the filters are so short that discrete Fourier transform techniques do not offer a significant advantage; indeed, due to the additional overhead that would be involved with respect to the algorithm which is to be presented, and since complex, floating-point arithmetic would be required, it is highly doubtful that there would be any advantage to this approach.

Finally, in many cases, all we want to see is a few frames of the output. Using a space-time implementation, this is straightforward; we just compute the desired output frames. However, one cannot do this in the frequency domain; rather one must compute the entire discrete Fourier transform of the input image in order to compute even one point of the filtered output.

3.1.2 Constant-velocity Theorem for Discrete Case

For those fortunate enough to have special hardware, such as array processors or pipelined fast Fourier transformers, the previous argument may carry little weight; the difference between one milliseconds and one hundred milliseconds of computation time is often easy to overlook. However, for those persons, we have a more convincing argument; the nature of the discrete Fourier transform is such that the nice geometry of the Fourier transform of continuous-domain constant-velocity functions does not appear in the discrete domain. In particular, we present here the closest thing to a discrete version of the constant-velocity theorem, which is derived in Appendix C:
Theorem 3.1.1

If \( x[m] \) is a one-dimensional discrete function, defined for \( m = 0, 1, \ldots (M-1) \), and \( y[m, n] \) is a two-dimensional discrete function which can be expressed as a constant-velocity mod-M rotation of \( x[m] \), so that \( y[m, n] = x[(m - vn) \mod M] \), for \( n = 0, 1, \ldots (N-1) \), where \( v \) a constant, then the two-dimensional discrete Fourier transform of \( y[m, n] \) is given by:

\[
Y[k, l] = \begin{cases} 
\frac{1}{N} X[k] & \text{when } \left( \frac{k}{N} + \frac{(kv) \mod M}{M} \right) \text{ is an integer} \\
\frac{X[k]}{MN} \left[ 1 - \exp \left\{ -j2\pi M \left( \frac{k}{N} + \frac{(kv) \mod M}{M} \right) \right\} \right]^{-1} & \text{else}
\end{cases}
\]

where \( X[k] \) is the discrete Fourier transform of the function \( x[m] \), and \( k = 0, 1, \ldots (M-1) \) and \( l = 0, 1, \ldots (N-1) \).

This result is discussed in detail in Appendix C, however we will make some important observations here. Clearly, this is not a simple extension of the two-dimensional continuous result, although, as shown in Appendix C, there are special cases in which the continuous and discrete results are similar. This result illustrates nicely the aspects of the frequency-domain approach which make it unattractive in this application.

First, we note that the input \( y[m, n] \) is finite in both space and time; this clearly must be so if the computer is going to process the function. Secondly, rather than representing constant-velocity translation, the input has the form of a constant-velocity rotation; because of the mod-M operator, each "row" of \( y[m, n] \) is found by rotating \( x[m] \) by \( vn \) elements. In other words, as \( x[m] \) translates, the elements at one end of the image "wrap around" to the other end. This same wrapping effect appears in the expression for the discrete Fourier transform, \( Y[k, l] \). This wrapping is due to the
underlying periodicity in the computation of the discrete Fourier transform as described in Appendix C, and also in [60]. A typical discrete Fourier transform pair is shown in Figure 3.1.1, and Figure 3.1.2.

Finally, we note that the term in square brackets has the same form as the discrete Fourier transform of a rectangular pulse; in that sense it is the discrete analog to the sinc function in the continuous domain. Thus, there is "ringing" around the line \( \left( \frac{k}{N} + \frac{(kv + m\text{mod}M)}{M} \right) = 0 \). This looks somewhat familiar; it strongly resembles the result we obtained in the case of temporal windowing, and in fact that is where it comes from. In order to compute the discrete Fourier transform of \( y[m, n] \), there is an implicit assumption that the function has finite extent. Even if we had a discrete function which represented constant-velocity translation for all time, we would not be able to compute its discrete Fourier transform. The best we can do is to multiply that function by a rectangular window, and then find the discrete Fourier transform of the resulting finite-extent function. The effects of this temporal windowing are evident in the discrete Fourier transform, as they would be in the continuous case.

As pointed out in Appendix C, there are special cases where this "ringing" goes away, leaving us with a discrete impulse line in the discrete Fourier transform. These special cases arise where the constant-velocity segments in \( y[m, n] \) are aligned with the constant-velocity segments in the neighboring copies of \( y[m, n] \) that are implicit in the discrete Fourier transform formulation. An analogous situation in the continuous case arises when constant-velocity segments of the same velocity are aligned so as to be indistinguishable from a constant-velocity function for all time; then the overall Fourier transform must be indistinguishable from the transform of the constant-velocity function, i.e. the ringing contributed by each constant-velocity segment must cancel everywhere except along the constant-velocity line, i.e. \( \omega_x + \nu \omega_y = 0 \).

3.2 Spatio-temporal Approximation Algorithm

Having decided to implement the velocity-tuned filter in the space-time domain, we now present the basic algorithm. Let the input image be \( I[x, y, t] \), where \( x \), \( y \) and \( t \) are discrete variables. An alternative view is that the input is like a movie, consisting of a sequence of spatial frames, \( I_t[x, y] \). We will call the three-dimensional input and output images image sequences, to avoid confusion with
Figure 3.1.1 Constant Velocity Mod·M Rotation of Discrete Function
Figure 3.1.2 Discrete Fourier Transform of Function in Figure 3.1.1

(Magnitude)
the two-dimensional images which comprise them. The filter is represented by \( f[t] \), the discrete form of the inverse Fourier transform of the filter generator function of the velocity-tuned filter.

In Section 2.2.6, we saw that the continuous domain velocity-tuned filter performs a one-dimensional convolution of each one-dimensional constant-velocity slice of the input image by the one-dimensional filter:

\[
g(x_0 + v_xt, y_0 + v_yt, t) = f(t) \ast_i (x_0 + v_xt, y_0 + v_yt, t)
\]

By analogy then, we will implement the discrete domain filter as a similar, discrete convolution:

\[
G[x_0 + t\Delta x, y_0 + t\Delta y, t] = f[t] \ast I[x_0 + t\Delta x, y_0 + t\Delta y, t]
\]

\[
= \sum_{t'=0}^{T_r-1} I[x_0 + (t - t')\Delta x, y_0 + (t - t')\Delta y, (t - t')f[t']
\]

One implementation approach would be to move through the three-dimensional input image, extract all the constant-velocity slices, perform the one-dimensional convolution, and store the one-dimensional result into the corresponding constant-velocity slice in the output image. However, there is a simpler way, with less overhead needed.

Letting:

\[
x = x_0 + t'\Delta x
\]

\[
y = y_0 + t'\Delta y
\]

then:

\[
G[x, y, t] = \sum_{t'=0}^{T_r-1} I[x - t'\Delta x, y - t'\Delta y, t - t']f[t']
\]

In words, each point in the output image is formed by computing a weighted sum of points in the input image. To compute the point at \([x, y, t]\) in the output image, we select the same point in
the input image, and the points which can be reached by adding multiples of \( \Delta x \) to its \( x \)-coordinate and \( \Delta y \) to its \( y \)-coordinate. These points are weighted by the values of the shifted filter function \( f[t] \). The filters we are interested in, such as the Gaussian, have a peak at zero, and decrease symmetrically from that peak. From the equation above, the filter value \( f[0] \) is used when \( t' = 0 \); the input point which it multiplies is \( I[x, y, t] \). Thus, the point in the output image at \( [x, y, t] \) is computed from the point in the input image at the same location, weighted most heavily, then the points located at \( [x \pm \Delta x, y \pm \Delta y, (t \pm 1)] \), weighted equally, but less than the central point, etc. This is typical behavior for interpolation and approximation filters.

When the velocity parameters \( \Delta x \) and \( \Delta y \) are integers, the constant-velocity slice "goes through" individual pixels in each frame \( I_t[x, y] \), so there is no ambiguity about what the filter values should be multiplying. On the other hand, when the velocity has non-integer values, the constant-velocity slice passes "in between" pixels in at least some of the frames. In this case, one can perform a weighted sum of nearby pixels to interpolate the value at this intermediate point; this weighted sum is then multiplied by the filter value for that frame, as if it were an actual pixel.

There is another way of viewing this algorithm, which lends light to the whole spatio-temporal approximation task. In effect, what this algorithm is doing is calculating a weighted sum of the image frames themselves. More precisely, each image frame in the output is formed by shifting the input frames by \( \Delta x \) and \( \Delta y \) relative to each other, multiplying the frames by the appropriate filter value, and adding them all up. The next frame in the output sequence is formed the same way, except that the filter values are all shifted in time by one pixel. This view of things is equally valid in the continuous domain; the shifting operation comes about from convolving the input with the mysterious impulse line function. This interpretation of the algorithm may also make it more clear why the velocity-tuned filter works the way it does. When an input function is a constant-velocity translation of some function \( f[x, y] \), then each frame in the discrete image, (or each time-slice in the continuous image), is the same as any other, but is offset in space by an amount which is determined by the velocity parameters. A velocity-tuned filter which is tuned to the same velocity as the function shifts the frames back by the same amount, so that the original images line up exactly; then the weighted sum of the frames
Section 3.3.1  

Examples of Two-dimensional Spatio-temporal Approximation

just returns a scaled copy of this image, with no other change. When the filter is tuned to the wrong velocity, then the original image is added to copies of itself which are spatially offset. The result is exactly the same as passing the image through a spatial filter, which would compute a weighted sum of each pixel in the input frame with the pixels around it; the effect is a spatial smearing, assuming that the filter in question is a low-pass filter.

There is one final note about implementation. In Chapter 2, we made a distinction between the effect of velocity-tuned filters on continuous images, vs. their application as reconstruction filters for sampled images. In the discrete domain, there is no such distinction; in a sense, all images are sampled. However, we would like to examine these two important cases, so we make the following approximation. “Continuous” images and image sequences will be those for which all pixels in all frames can be non-zero. Conversely “sampled” images will be zero for all times, except at predetermined “sampling times”; typically, a sampled image will be made by multiplying a continuous image by a mask which has unity value at the sampling times, and zero otherwise. In the examples to be presented, the sampling was done in a periodic fashion, but it would not need to be, in principle, to demonstrate the results.

3.3 Examples of Two-dimensional Spatio-temporal Approximation

We now present the results of several experiments which demonstrate the properties of two- and three-dimensional velocity-tuned filters, both in the context of filtering continuous images, and as spatio-temporal approximation filters. When viewing the pictures presented in this section, it is assumed that the bound edge of the picture is the “top” edge; the grey-scale color bar should be at the right-hand side of the picture.
3.3.1 Constant-velocity Input

In Plate 1, we demonstrate the properties of the two-dimensional velocity-tuned filter. A legend for Plates 1 and 2 appears in Figure 3.3.1. The left group of images (images 1-9) depicts the effects of a velocity-tuned filter on a "continuous" input; the right group (images 10-18) is the same as the left, but the input has been "sampled" first. Within each image, the horizontal axis represents the spatial dimension, and the vertical axis the temporal dimension.

The one-dimensional function \( h[x] \) which is translating is coming up out of the page; in effect, we are looking down at the top of it. In this case, \( h[x] \) is just a rectangular pulse. It is moving with a constant velocity so that the constant-velocity function \( g[x, t] = h[x - t \Delta x] \) is a rectangular bar lying in the image; the velocity is equal to the slope of the bar in the image. The velocity of the input is the same in all images, and has value \( \Delta x = 1 \).

Within each group of images in Plate 1, we have varied the velocity and bandwidth of the velocity-tuned filter used. In the space-time domain, the filter used was a discrete Gaussian filter, whose size was controlled by the parameter \( \sigma \). In the frequency domain, this corresponds to using a velocity-tuned filter whose filter generator function is also a Gaussian, and whose bandwidth varies as \( \sigma^{-1} \). In the examples shown, the filter velocities used were \( \Delta x = 1 \), \( \Delta x = 2 \), and \( \Delta x = 4 \), and had \( \sigma \) values \( \sigma = 1 \), \( \sigma = 2 \) and \( \sigma = 4 \).

The important points demonstrated by the left group of images are as follows. When the velocity of the filter and the input are the same (image 1), then the input is passed unchanged by the filter. As the velocity of the filter and of the input become more disparate, the output of the filter continues to be a constant-velocity function of the same velocity as the input, but the bar is spatially low-pass filtered, or smoothed by the filter. Further, for a given velocity disparity, as the value of \( \sigma \) increases, the smoothing effect becomes more pronounced. This would seem obvious from the description of the spatio-temporal algorithm above, but it demonstrates a more subtle concept from the frequency domain. As \( \sigma \) increases, the bandwidth of the filter generator function decreases, thus decreasing the effective velocity bandwidth of the velocity-tuned filter; as shown in Chapter 2, this leads to a more
Figure 3.3.1 Legend for Plates 1 and 2
prominent low-pass filtering effect of constant-velocity inputs whose velocity is different than that of the filter.

### 3.3.2 Sampled Constant-velocity Inputs

The experiments which produced the left group of images in Plate 1 were repeated exactly to produce the right group of images, except that we have "sampled" the input function. The remarks about the previous case are still applicable, but now we add some new ones. First, when the velocity of the filter is the same as the velocity of the input, and when the $\sigma$ value is large enough, then an exact reconstruction of the original constant-velocity function is obtained from its samples; this is shown in the lower left-hand image (image 16) of the group. If the $\sigma$ value bandwidth is too small, the reconstruction is only partially successful, as shown in the images immediately above it (images 10 and 13).

In the space-time domain, the $\sigma$ value must be large enough that the spatio-temporal filter can "connect the dots", as discussed in Section 2.3, if exact reconstruction is to occur. In the frequency domain, increasing $\sigma$ decreases the bandwidth of the filter, and makes it more effective in eliminating the spectral replicates in the Fourier transform of the sampled function; exact reconstruction is possible only when all these replicates are eliminated.

When the velocity of the filter is not the same as the velocity of the original function, an interesting thing happens; the samples are smeared in the wrong direction, so that exact reconstruction does not occur; this is shown best by the image in the lower right-hand corner (image 18). If the velocity of the filter is close enough to the velocity of the original function, then an approximate reconstruction can occur; there is spatial smearing as in the continuous case. As the velocities become more disparate, the reconstruction becomes worse, but the smearing effect diminishes; this is demonstrated, though somewhat obscurely, by the lower-right image (image 18). If we examine a horizontal slice of the image, i.e. look at a temporal slice of the image, we would see more than one rectangular pulse, indicating that the reconstruction failed; however, each rectangular pulse is an unsmeared version of the original
image. This point will be made much more clearly in the three-dimensional case, so one shouldn't strain oneself trying to see it here.

3.3.3 Non-constant-velocity Input

The experiments which were done to prepare Plate 1 were repeated exactly to prepare Plate 2, except that a parabolic translation function, as described in Section 2.5 was used to control the trajectory of the one-dimensional rectangular pulse, \( h(x) \); thus \( h(x) \) undergoes rigid-body translation, but at a non-constant velocity.

The left group of images were generated by passing a continuous input image through a set of velocity-tuned filters, and demonstrates the following points. Clearly, only a portion of each trajectory will have the same velocity, \( i.e. \) slope, as the velocity-tuned filter which is used; such portions are passed unchanged by the filter. In addition, a surrounding region of the trajectory may have velocity which is "close enough" to the velocity of the filter to be passed nearly unchanged; the concept of effective velocity bandwidth is demonstrated by observing how large these regions are. As expected, we see that as the bandwidth of the filter decreases, \( i.e. \) as \( \sigma \) increases, these regions become smaller; moreover, in that part of the trajectory which lies outside these regions, the spatial smearing effect of the filter becomes more pronounced. In the bottom row (images 7-9), where the velocity bandwidth is smallest, we see this effect most clearly.

In the right-hand group of images, we have sampled the input as before. Here we see that only those regions which have approximately the same velocity as the filter are reconstructed correctly. Outside these regions, we see the effect described in Plate 1; the samples are smeared in the direction determined by the filter velocity rather than the velocity of the original function, so that reconstruction does not occur. In addition, as in Plate 1, the bandwidth of the filter must be small enough to eliminate the spectral replicates from the sampled function, in order for reconstruction to occur; only in the bottom row (images 16-18), where \( \sigma \) takes on its maximum value, does this occur.
3.4 Examples of Three-dimensional Spatio-temporal Approximation

Admittedly, the two-dimensional images may be a little hard to comprehend, especially upon first encounter. However, they do demonstrate the essential properties of velocity-tuned filter and especially in the context of reconstruction of sampled images. We now demonstrate these points again, in the three-dimensional case. The next group of plates depict the rigid-body translation of a two-dimensional image. In an attempt to make our points more clear, we have used an something that one might be likely to observe engaged in rigid-body translation, namely a Ferrari 308/365GTB; hopefully, this is easier to envision than a translating rectangular pulse. In other words, in the following experiments, the input to our three-dimensional velocity-tuned filter will be a sequence of images which represent the constant-velocity translation of the original Ferrari image. In the plates that follow, we present representative frames from the image sequences that are the output of the three-dimensional velocity-tuned filter, rather than presenting the whole image sequence.

3.4.1 Constant-velocity Inputs

In Plate 3, we show the effects of the velocity-tuned filter on continuous images. In contrast to the two-dimensional case, we have translated the Ferrari at differing velocities, and used the same velocity-tuned filter for all experiments. In the examples shown, the velocity of the filter was \( \Delta x_f = 1 \) and \( \Delta y_f = 0 \). In each case, the image translation was purely horizontal, so \( \Delta y = 0 \); the horizontal velocity components used were \( \Delta x = 1, \Delta x = 3, \) and \( \Delta x = 5 \).

As before, the one-dimensional filter \( f[t] \) which was used in the spatio-temporal convolution was a discrete Gaussian filter, whose size was controlled by \( \sigma \); thus the effective velocity bandwidth varied inversely as \( \sigma \). Two \( \sigma \) values were used, \( \sigma = 2 \) and \( \sigma = 4 \).

Each image shown is a representative frame from the output image sequence of the velocity-tuned filter in response to a particular input sequence. Within each row, the velocity of the input image is the same; the left-hand image was produced by a filter with \( \sigma = 2 \); for right-hand image, \( \sigma = 4 \). In the top row of images, the velocity at which the original image was translating is the same as the filter
velocity. In subsequent rows, the velocity of the input image increases; thus the disparity between the filter velocity and the input velocity also increases.

We wish to emphasize again the following points. When the velocity of the filter is that same as the velocity of the translating image, then the image sequence is passed without any change. There is absolutely no spatial smearing, so there is no loss of spatial resolution. As the difference between the velocity of the filter and the velocity of the image increases, then smearing appears, in the direction of the velocity of the filter; in this case, the filter velocity components are $\Delta x_f = 1$ and $\Delta y_f = 0$, so the smearing is horizontal. As in the two-dimensional case, the extent of the smearing increases as the value of $\sigma$ increases. The smearing is actually spatial smearing of the input image; by squinting or viewing these images at a distance, one sees that their appearance becomes more similar.

As in the two-dimensional case, these results can be interpreted in the frequency domain as follows. As $\sigma$ increases, the bandwidth of the Gaussian filter generator function for the velocity-tuned filter decreases, thus decreasing the effective velocity bandwidth of the velocity-tuned filter. As this happens, smearing becomes significant for smaller disparities between the filter and input velocities; in addition, the smearing effect becomes more pronounced.

### 3.4.2 Velocity-Tuned Reconstruction

To produce Plate 4, we "sampled" the image sequence which was used as the input in the previous case; the sampled input sequence was entirely zero, except for every fifth frame, which was a copy of the corresponding frame of the "continuous" input. The sampled sequence was passed through a three-dimensional velocity-tuned spatio-temporal approximation filter. As before, the filter function used, $f[t]$, was a discrete Gaussian. The two groups of images shown in Plate 4 are five successive frames of the image sequence output by the filter. The only difference between the groups is the $\sigma$ value of one-dimensional filter; the left-hand group was generated with $\sigma = 2$, whereas for the right-hand group, $\sigma = 4$. 

153
From the images we see that the right-hand group is a correct reconstruction of the original input sequence, but the left-hand group is not. In the spatiotemporal domain, our interpretation is that the spatiotemporal extent of the filter used for the left-hand group was not large enough to smear the samples sufficiently to "connect the dots" as discussed earlier. Our frequency domain interpretation, as usual, is that the effective velocity bandwidth of the velocity-tuned filter used to generate the right-hand group was small enough to eliminate the spectral replicates from the spectrum of the sampled function, whereas the effective velocity bandwidth for the left-hand group was not.

We note that in the case of incomplete reconstruction, there is no spatial smearing; the spatial information in each image is correct, but the overall intensity of the output varies from frame to frame, whereas it is constant in the correctly reconstructed sequence. Consequently, when the right-hand sequence is viewed as a movie, one sees the correctly reconstructed constant-velocity motion of the original Ferrari image. When the left-hand sequence is viewed as a movie, one still sees the correct motion of the Ferrari image, and the spatial detail is preserved, even though the reconstruction was not correct; however, flicker is observed in this case, due to the varying intensity in the image sequence.

Plate 5 demonstrates the effect of velocity-tuning on spatio-temporal approximation. Each image in Plate 5 is a representative frame from the image sequence output by the velocity-tuned filter. In each case, the input sequence was a sampled sequence, depicting constant-velocity motion of the Ferrari image, as in Plate 4. As mentioned, the input sequence used to generate Plate 4 was zero, except for every fifth frame; for the following experiments, this "sampling interval" was decreased to four frames. The filter used was again a discrete Gaussian filter; it was tuned in each case to the same velocity: \( \Delta x_f = 1 \) and \( \Delta y_f = 0 \).

Within each row of Plate 5, the velocity of the car is constant, and the \( \sigma \) value of the filter is changed. The left-hand images were generated by a filter with \( \sigma = 2 \); for the right-hand images, \( \sigma = 4 \). Both \( \sigma \) values are sufficiently large to obtain reconstruction of the input sequence. In the top row, the velocity of the filter equals the velocity of the input; in subsequent rows, the velocity of the input sequence is increased, while that of the filter remains constant; thus each column represents increasing
velocity disparity. The input velocities used were all horizontal, and had $\Delta x = 1$, $\Delta x = 2$, $\Delta x = 3$, and $\Delta x = 5$.

In the top row, the filters used are tuned to the same velocity as the original function, so a correct reconstruction is performed. In the lower rows, the velocity of the filter diverges from the velocity of the Ferrari, so the reconstruction is not exactly correct. In the second row, the reconstruction is still pretty good; some smearing is evident, especially in the right-hand image, but it is not too annoying. The lower images might be best described by saying that they look as if multiple-exposures were made of the moving car. This case is analogous to the lower-right hand image in Plate 1. At each time instant, we see multiple copies of the original image. The spacing between each of the copies increases as the velocity disparity increases; the number of copies which are visible is larger in the right-hand group because the $\sigma$ value of $f[t]$ is larger, thus increasing the spatiotemporal extent of the filter. We note however, that when the copies are sufficiently separated spatially, the image is not spatially smeared, as it was in the continuous case depicted in Plate 3; indeed, one could argue that the details of the image are more clearly discernible in each of the copies than they were in the continuous case.

In the three previous plates, the velocity of the filters used were in the same direction as the velocity of the car. Just to make these results a little more general, the experiments were repeated with filters whose velocities had different magnitude and direction than the input image sequence. The images in Plate 6 are were produced with the same input sequences as the preceding plates. They represent the results of filtering both the "continuous" and "sampled" case, where the filter velocity has both horizontal and vertical components of equal magnitude; in particular, $\Delta x_f = 1$ and $\Delta y_f = 1$. We see that the results are predictable extensions of the previous cases. It does seem like the distortion introduced by the filters is a little greater in these cases, but that may well be because we are more used to seeing fast cars blurred horizontally than at a forty-five degree angle.

The three-dimensional results may be made fairly intuitive by reconsidering the concept of computing weighted sums of shifted input images, as discussed above. In the "continuous" case, there are a relatively large number of frames added together, and the relative shift between neighboring frames can be small, on the order of one or two pixels; this can produce a relatively large spatial smearing effect.
In the "sampled" case, a relatively smaller number of non-zero images are included in the weighted sum, and the relative shift between the images is larger, on the order of five to fifteen pixels in these examples. As a result, the perception is not so much one of spatial smearing as it is of "multiple exposure"; to be sure, the output sequence differs greatly from the original, unsampled image sequence, however, much of the spatial details in the image frames is still discernible.

This phenomenon was predicted on theoretical grounds by Fahle and Poggio [36], in reference to vernier acuity experiments. If an experimental subject is asked to distinguish a left-oriented vernier from a right-oriented one, it seems unlikely that perfect reconstruction is required. Even in the cases where reconstruction is poor, the spatial detail may still be present to a large extent, and would be available to perform a discrimination task such as the vernier acuity experiment. As detailed by Fahle and Poggio, this can be explained by elegant frequency domain arguments, regarding the nature of the spectral replicates when an function is temporally sampled; they present a similar analysis which shows that this is not true in the case of a spatially sampled input, such as was used in Burr's vernier acuity experiments [35]. From their analysis, and from our results, one could argue that the ability to perform the vernier hyperacuity tasks for temporally sampled moving verniers is only weak evidence for velocity-tuned spatio-temporal approximation in the human visual system. However, other psychophysical results, such as apparent motion, and the reconstruction of spatially sampled verniers still remain as strong evidence.

3.5 Objects Moving in the Presence of Noise

In the next chapter, we will discuss a fairly interesting prediction, based on the spatiotemporal description of the velocity-tuned filter. Specifically, if an object is moving at constant velocity in the presence of noise, and if the resulting image is filtered by a velocity-tuned filter tuned to the velocity of the object, one can show that the filter increases the detectability of the object. In essence, the filter passes the image of the object unchanged, but decreases the variance of the noise by smoothing it along constant-velocity lines. This will be discussed in more detail in Chapter 5, and the effect will be verified in humans through psychophysical experiments.
In Plate 7, we have shown the results of simulating the psychophysical experiment of Chapter 4. In the top row, we see the entire input image. It is a sequence of five frames; the length is intentionally short for reasons which pertain to the psychophysical experiment. The sequence depicts a "C"-shaped object moving at constant velocity, with \( \Delta x = 4 \) and \( \Delta y = 0 \). In each case, the image sequence was filtered with a velocity-tuned filter; once again, a discrete Gaussian filter was used, with \( \sigma = 4 \).

The second row depicts the result of filtering the input with a velocity-tuned filter tuned to the velocity of the object. To produce the third row, the same filter was used, but the velocity of the object was changed to \( \Delta x = 0 \). To generate the final row, the velocity of the object was restored to \( \Delta x = 4 \), but the order of the frames was "randomized", as they will be for the psychophysical experiment.

We emphasize the following results. When the filter is tuned to the velocity of the object, the object is passed unchanged, as predicted. In addition, the variance of the noise is decreased, making the background look more uniform, and making the object more clearly visible, as seen in the second row. In the third and fourth rows, we get the same amount of noise reduction. However, in the third row the object is smeared by the filter, due to the disparity between the velocity of the filter and of the object. When the order of the frames is randomized, the smearing effect is even worse, as would be expected on the basis of the spatiotemporal description of the velocity-tuned filter. In the third row, the direction of the "C" is pretty clear, but probably not much more than in the original image. In the last row, the smearing is so severe, that the orientation of the character is quite unclear. We note that since the velocity-tuned filter smears horizontally, the top and bottom horizontal segments of the character appear less smeared than the vertical character segments.

In the psychophysical experiment of Chapter 4, the subject's task will be to discriminate a left-facing "C" from a right-facing one. One could predict from our results in this chapter that when the "C" moves at, or reasonably near the velocity of a velocity-tuned filter, then it should be easier to perform this task. On the other hand, if the velocity of the object is too different from that of the filter, the object will be smeared; even though the filter will still act to decrease the noise level, the smearing may result in an overall decreased ability to discriminate the two shapes, relative to the unfiltered sequence. The ability to discriminate the shapes will certainly be higher in the case where the velocity of the object
and the velocity of the filter are nearly the same, then when the velocities are very different. In the extreme case of a randomized input sequence, where the velocity of the object is constantly changing, the object can be severely smeared, so the ability to discriminate the shapes should be degraded. These predictions will be examined in the context of psychophysical experiments in the next chapter.
Examples of Velocity-tuned Filtering and Velocity-tuned Reconstruction

Computational theories are nice, but are always questionable in the absence of experimental evidence which support them. In this chapter, we present a specific prediction which can be inferred from the spatio-temporal approximation formulation, and present experimental verification of that prediction.

4.1 Spatio-temporal Approximation and Noise

In a few instances, we have noted that any signal-processing system has to be designed to work in the presence of noisy signals. This is certainly true for vision systems, where the signal that is being measured, i.e. light intensity, is itself governed by the probabilistic nature of photon emission and detection, and is thus guaranteed to be noisy. In practice, this may not be a significant noise source; photon emission is well-modeled as a Poisson process, so that the signal-to-noise ratio in an intensity measurement increases as the square-root of the signal level. In situations where there is a low photon emission rate, or where the detector only captures a small fraction of the photons which hit it, the signal-to-noise ratio may be very poor. In addition there are the usual numerous opportunities for the injection of noise after the detection stage, either in amplifiers and quantizers, or neurons.
Without delving into too much signal detection and estimation theory, let us merely state the following. Many types of noisy signals can be broken up into a signal component and a noise component. Smoothing or averaging noise tends to decrease its power. In the time or space domains, this is because local fluctuations in the noise tend to be canceled out by nearby fluctuations of the opposite sign. The result is a noise signal which has a smaller variance; its value tends to change less rapidly over a given interval than that of the original signal. This assumes, of course, that the noise is not already smoother, i.e. more correlated, than the smoothing operator which is to be applied. In the frequency domain, smoothing implies low-pass filtering, i.e. attenuation of the higher frequency components of the signal. If the noise contains frequency components which lie outside the passband of the low-pass filter, these components will be removed, thus decreasing the overall power in the noise. When this is not the case, the noise is already smoother than the operator, so the smoothing has no effect.

Now, if the smoothing operation either leaves the signal component of the noisy signal unchanged, while decreasing the noise power, then smoothing the signal-plus-noise combination will increase the signal-to-noise ratio, which in turn increases one’s ability to detect the signal in the noise, i.e. to distinguish the noisy signal from noise alone.

Let us now consider the case of a velocity-tuned filter whose input is a constant-velocity function, corrupted with uncorrelated noise. From the discussion in previous sections, we know that the velocity-tuned filter will pass the constant-velocity function nearly unchanged as long as the velocity of the filter is “close enough” to the velocity of the input; as these two velocities become more disparate, the input gets smeared in the spatial dimension to an extent which depends on the bandwidth of the filter and the original function. We also know that the velocity-tuned filter is a low-pass filter, or smoothing operator; thus the power in the noise component in this noisy signal will be reduced by the filter, as described above. Thus passing the noisy signal through a velocity-tuned filter should increase the signal-to-noise ratio, thus increasing the detectability of the constant-velocity function.

Now, in the case where the input is a randomly moving function, corrupted with uncorrelated noise, the situation is different. The noise power is reduced as before, but the signal component is
highly degraded. As we saw earlier, a velocity-tuned filter will tend to smear such an input along lines of constant velocity in the space-time domain. As a result, the signal-to-noise ratio in this case will not increase as much as it did in the previous case. Thus, the randomly moving function should be less detectable in the presence of noise than the one moving at constant velocity.

These same arguments can be extended to the case where the input is a moving function, with uncorrelated noise, which has been sampled. Of course, the best results should occur when the velocity-tuned filter is able to reconstruct the original function from its samples, and we have already discussed the conditions under which this will happen, in Section 2.3. In the case where the input is randomly moving, the reconstruction will be clearly be poorer than in the constant velocity case, so we may expect a larger disparity in the signal-to-noise ratio of the constant-velocity case relative to the randomly-moving case. However, often reconstruction does not have be very good if detecting a function only requires discriminating it from a small set of possible functions; this is true in the case of temporal sampling and the vernier hyperacuity task, as discussed in Chapter 1. Thus the effect of poor reconstruction is in general rather hard to predict quantitatively.

In view of these arguments, we can make the following prediction. If velocity-tuned filters such as we have discussed are used by the human visual system in performing the spatio-temporal approximation task, then the detection and identification of objects which are moving at constant velocity in the presence of noise should be superior to that for objects which are moving randomly. The intent of this experiment is to verify this prediction.

4.2 Methods

One approach would be to design an experiment to determine the threshold contrast-sensitivity for an image moving at constant velocity in the presence of noise, and for the same image moving randomly, and to compare the results. However, it is the comparison that we are really after, so the goal of this experiment is to expose a difference in threshold contrast-sensitivity between the two cases. To achieve this goal, the experiment was designed to simultaneously seek (but not measure quantitatively)
the threshold contrast-sensitivity level for both types of stimuli, using a modified staircase approach, and compare the trajectories.

Each experiment consisted of 200 trials. On each trial a short "movie" was presented to the subject, who was then asked for a two-alternative forced-choice response. Each movie, or image sequence consisted of five frames, and was presented at a frame rate of thirty frames per second, on a Symbolics raster-scan video display. Thus, the total duration of the presentation was about 165 milliseconds; durations this short should preclude eye pursuit movements, [34]. Each frame in the image sequence was 64 by 64 pixels, with each pixel containing 8 bits, allowing 256 levels of grey intensity. The image sequence was viewed in a dimly-lit room at a distance of one meter.

Each frame of the movie was the sum of a signal image and a noise image. The signal image consisted of a synthetic geometric shape, of amplitude $A$ on a background level of 128, (the middle grey level on the CRT). The geometric shapes chosen for the signal images were that of a capital "C" and of its left-right mirror image. On any particular trial, one of four possible image sequences were created. In the first type, each frame contained the right-facing "C" shape, displaced by a constant displacement, $\Delta x$ to the right of its position in the previous frame. When displayed as a movie, this sequence appeared to show the character moving at constant velocity to the right. The second image sequence was the same as the first, except that the left-facing "C" was used. The third image sequence consisted of exactly the same frames as the first sequence, except that the order of the frames was randomized. When displayed as a movie, this sequence appeared to show a "C", moving about randomly. The fourth sequence was the same as the second sequence, only randomized in the same manner as the third.

The relative displacement of the "C" between frames of the constant velocity stimulus was 4 pixels in each case. This represents a distance of about 2.42 mm. This displacement occurs in 1/30 (temporal) second; at a viewing distance of 1 meter, this translates to an angular velocity of about 4.2° per second. At this velocity, the apparent motion effect is fairly strong, suggesting, within our theoretical framework, that the velocity of the stimulus is within the velocity bandwidth of a velocity-tuned filter.
Each noise image was an approximation to uncorrelated Gaussian noise, with zero mean and standard deviation $\sigma = 30$, created by summing the output of a uniformly-distributed pseudo-random number generator. The noise in the noise image sequence was uncorrelated not only within each frame, but also from frame-to-frame, i.e. both spatially and temporally. When viewed as a movie, such a sequence look like the "snow" pattern on an untuned television set. A new noise image sequence was generated for each trial.

As mentioned, the goal of the experiment was to seek the threshold contrast-sensitivity level for each type of stimulus; i.e. to find the lowest contrast level at which the observer could reliably detect the signal in each case. This was under the control of a psychophysical experiment management system, developed by the author on the Symbolics Lisp Machine for this purpose.

On each trial, one of the four types of signal image sequences was chosen at random. If an image sequence of this type was not available at the desired contrast level, a new one was created. A noise sequence was generated and added to the signal to create the test sequence; the subject was then notified that the computer was ready to proceed. The subject clicked a button on the Lisp Machine mouse, to indicate that he or she was ready; the computer responded by showing the test sequence, as a movie. The computer then queried the used for a response by exposing a graphics display on the screen; the subject was asked to indicate whether the character in the movie opened to the left or right by pointing to the appropriate side of the display with the Lisp Machine mouse and clicking a button.

As is usual in two-alternative forced-choice experiments, it is assumed that the subject is detecting the signal if the responses are correct 75% of the time or more; 50% correct can be achieved by guessing randomly, and 100% correct indicates detection with very high probability. The job of the computer then is to adjust the contrast of each stimulus type until the subject responds correctly 75% of the time. Thus as the experiment proceeded, the computer tracked the results for each of the two stimulus types, i.e. constant-velocity and random-motion. After a stimulus type had been presented at a given contrast level for a preset number of times, typically 8, the computer computed the percent-correct for that stimulus-type and contrast setting, and adjusted the contrast for that type of stimulus up or down so that the percent-correct at the new contrast would be driven towards 75%, and proceeded to the next
trial. The computer recorded the results of all trials, grouped by stimulus type and contrast level. After the last trial, a graph of contrast level as a function of trial number, i.e. the contrast trajectory, was generated.

This staircase approach was modified as follows. At the beginning of the experiment, the contrast level of both stimulus types was set to the same level, high enough that the observer could easily distinguish the left-facing from right-facing characters. If the user responded correctly, he or she was offered the opportunity to decrease the contrast level for that stimulus type immediately, rather than performing the usual eight trials first. This option was repeated as long as the user continued to respond correctly; after the first wrong answer, the staircase procedure was started. The amount of the decrease on each of these steps was the same as it would have been if eight trials had been performed with 100% correct response level; thus the intent was to allow the user to decrease the contrast rapidly to the vicinity of the threshold-contrast, and then start the staircase procedure.

In addition, in some experiments, if during the course of the staircase procedure, the subject obtained six consecutive correct responses, the offer to decrease the contrast was again extended. The justification here was that this situation would arise only if the contrast was clearly above the threshold contrast, or if the user was very lucky. In the first case, it would be desirable to allow the user to decrease the contrast quickly towards the threshold. In the second case, i.e. where the subject is guessing blindly, but correctly, decreasing the contrast would only confound the process of finding the threshold-contrast; thus the subjects were carefully instructed to decrease the contrast only when they were sure of their responses, and felt confident that they were operating above their true threshold.

The motivation for these modifications is as follows. First, there is a definite training period needed for naive subjects. Persons such as the author, who have performed the test many times can generally attain lower threshold contrasts than untrained subjects. However, it is more important to demonstrate the effect in the naive subject than the experienced ones; after all, we started off asserting that the effects we are looking for are low-level and thus in a sense "automatic". Experienced subjects probably develop strategies that potentiate better detection, but it is not strategies that we are looking for in
this experiment. Thus, it is important to let the new subject find his or her own threshold contrast level. Unfortunately, even though each trial takes less than ten seconds, finding this level using the unmodified staircase procedure would be long, tedious, and tiring, and is thus to be avoided.

Truth be known, the whole experiment tends to get frustrating for subjects because half of the time, the contrast level is so low that they are looking at noise and then forced to respond "left" or "right". Each experiment ran about 45 minutes, and this was found to be a long time to perform a task such as this which requires focused concentration. Although the subjects were encouraged to take short breaks every ten minutes or so, it became evident that the shorter the test, the better. Yet one can only feel safe in concluding that the subject has reached threshold contrast when the contrast level starts to fluctuate above and below some constant level; i.e. enough data must be taken to allow the contrast to "equilibrate". Thus, the justification for allowing the user to decrease the contrast rapidly in the middle of the staircase was to help speed up this equilibration process, and get as much information as possible from a relatively small number of trials. These modifications introduce an element of risk inasmuch as they admit subjective judgment from the viewer to influence the success of the experiment. However, it was found that this judgment could be trusted in general, and felt that the tradeoff was acceptable.

Finally, a few things should be said about the signal image sequences used here. First, as already mentioned, all the channels in the human visual system are temporally bandlimited, and roughly to the same frequencies. As a result, any noisy image sequence looks less noisy when displayed as a movie, then when displayed statically; that is, the visual system temporally averages the noise in the image sequence, decreasing the noise variance, so the movie looks less noisy. The magnitude of this effect depends on the number of different frames shown (i.e. the number of "observations") during the integration interval, i.e. the temporal time-constant of the visual system. As a result of this effect, one thing which does not work is to compare a noisy movie with constant-velocity object to a noisy movie with an object that is not moving. The noise in the stationary case will be averaged to the same extent as in the moving case; thus one would expect at least the same degree of noise reduction in the two cases. By considering the stationary case as a constant-velocity case with zero velocity, it becomes clear that there should be little difference between these types of stimuli.
Secondly, we wish to explain the use of short image sequences in these experiments. One’s daily experience indicates that the natural tendency of the human visual system is to track something which is moving in the visual field. When the velocity of the object is constant and predictable, we seem to be quite good at it. Heuristically, one can argue that if the eyes track a moving object well enough, the image of the object will become nearly stationary on the retina. If that was to happen during these experiments, then we would essentially revert to the case of looking at a stationary object in the presence of noise; as we argued earlier, the detectability of the object should increase in this case, due to the temporal smoothing which is known to exist in the visual pathways. However, what we want to demonstrate is that spatiotemporal smoothing results in an increased detectability for objects which are moving across the retina, so tracking of the object is to be strenuously avoided here. We stress that this is a heuristic argument, and do not claim to know how well the tracking mechanisms actually work. However, by using presentation times which are short enough, we expect that the tracking mechanism will not have time to lock on to the object; indeed, experimental results [34] indicate that tracking eye movements require on the order of that 150 milliseconds to be initiated. Now, the display used for the experiments was a raster-scan CRT, refreshed 30 times per second; thus, we were forced to limit the length of the image sequences used to five frames, with a total presentation time of 165 milliseconds.

Finally, we note that it is very hard to “randomize” the order of five frames, and especially hard when the resulting image sequence is meant to represent random motion. If the set of frames is represented by the set of numbers \( \{1, 2, 3, 4, 5\} \), then such randomized results as \( \{5, 1, 2, 3, 4\} \) are unsatisfactory; they contain a large segment which represents constant-velocity motion. Even more random sequences such as \( \{1, 3, 5, 2, 4\} \) are not much better; this contains a constant-velocity segment, albeit of twice the nominal velocity. Therefore, it is important to limit the set of possible random sequences, to those which appear as random motion on the display. One way is to not admit a random sequence in which any sub-sequence of three frames has the same relative displacement. We went one step further, and allowed only sequences in which the displacement between adjacent frames switched signs, such as \( \{5, 2, 4, 1, 3\} \); the resulting sequences depict the object in the image “jumping back and forth” when displayed as a movie. The rationale here is that this is just the best way to maximize the relative displacement of the object in the image between frames, given the small number of frames. It does point out that velocity-tuned filters are direction-selective since the their support in the frequency
domain depends on the sign as well as the magnitude of the velocity of the filter. However, it should be noted that there is nothing special about switching directions; when the magnitude of the velocities are small, for instance, a filter tuned to $v_f$ can pass a constant-velocity function of velocity $-v_f$ with little change, depending on the relative bandwidths of the filter and the function.

4.3 Results and Discussion

The results of this experiment for four subjects are presented in Figures 4.4.1 through 4.4.5. The data for subject JGB are those of an experienced viewer; the remaining subjects were naive and the data shown represent their first and only exposure to the experimental setup.

Each figure is a graph which depicts the history of an experiment; the horizontal axis is the trial number. The vertical axis indicates the difference in grey-level intensity between the moving object and the mean value of the noise in the image. As mentioned, an 8-bit grey-level display was used, thus providing a dynamic range of 256 grey-levels. In all experiments, the noise had a mean grey-level of 128 units, and a standard deviation of 30 units. In a typical experiment, the initial grey-level intensity of the object was 50 units, and decreased over the course of the experiment to 10 or 20 units. We stress that these numbers represent grey-levels, and not physical intensity or contrast levels. In a more quantitative experiment, we would have generated a calibration curve to map grey-levels to actual intensity levels; indeed there are indications that the mapping is highly non-linear. However, we state once again that the purpose of the experiment is to demonstrate an effect, not to make a highly accurate measurement of its magnitude.

In each figure, there are two curves. As labeled, the solid curve represents the grey-level contrast of the random-motion stimulus used for a particular trial number; the dashed curve represents the contrast for the constant-velocity stimuli. In each experiment, the initial grey-level contrasts are high enough to guarantee that the subject can correctly perform the discrimination task. Over the course of the experiment, the contrast is decreased, under control of the computer, until the subject reaches the vicinity of his or her threshold contrast, and then fluctuates up and down, theoretically around that
Results and Discussion

200 trials
subject ET

Contrast vs. Trial Number
200 trials
subject jgb

Contrast vs. Trial Number
threshold. The result that we are after is the difference, if any, in the threshold contrast for the two stimulus types, i.e. randomly moving vs. constant velocity motion.

We start off with a summary statement; the results clearly demonstrate the predicted effect. In each case, the threshold contrast sensitivity is higher in the random-motion case than in the constant-velocity case. Other than that, there are few things that can be said that were consistently true for all subjects. Each subject had his or her own unique strategies for performing the experiment, and to some extent these are reflected in the contrast trajectories. Some subjects, like DTC, approached the threshold contrasts conservatively, decreasing the contrast only when completely sure of the response. Others, like SG, decreased the contrasts rather aggressively, tending to overshoot the threshold contrast somewhat, and then gradually rising towards the threshold later. In some cases, the contrasts tended to rise somewhat towards the end of the test, which is attributed to subject fatigue.

The nice thing about this experimental protocol is that since the two stimulus types are presented alternately (on average) in each experiment, each stimulus type acts as a control for the other. In other words, whatever forces are at work influencing the contrast trajectories affects both stimulus types at the same time. Any effect such as training, fatigue, or a strategy change which affects one trajectory tends to effect the other trajectory in the same manner. If the two stimulus types were presented in different experiments, at different times, the comparison of trajectories would be much less meaningful.

Previously, the observation was made that experiments which involve reconstruction of spatially sampled signals, such as Burr's, were computationally more difficult than those involving temporal sampling, such as Westheimer's, and that they provided stronger evidence for spatio-temporal approximation in the human visual system. The reader may wonder why then we chose to use temporally sampled signals for these experiments. The reason is that the goal of the experiment was to demonstrate the effect; the experiment chosen was "hard enough" to do this, yet had the advantage of being "easy enough" to implement and to expect untrained subjects to perform well. If it had turned out that this protocol did not reveal the phenomenon we had predicted, i.e. if there was no difference in detectability of the two stimulus types, it would certainly have been possible to make the task more challenging. For
example, it seems likely that presenting the subject with spatial samples of the constant velocity stimulus would lead to about the same threshold contrast; whereas presenting the spatial samples in random order would make the randomly-moving stimulus very hard to detect. This experiment is planned for the future, but is not needed here.

In spite of all these disclaimers about demonstration, rather than measurement of the effect, it seems fitting to close with the observation that, although the absolute numbers have a fair amount of variability, in each of the subjects, the ratio of the threshold contrast for the randomly moving stimulus vs. the constant velocity stimulus is roughly constant, being pretty consistently in the range of 1.2 to 1.4. Clearly, if we wanted to precisely measure this ratio, the experimental design would have to be changed. Having done so, it would also be interesting to see how this quantity varies with stimulus properties such as velocity and spatial frequency content for example, and for spatial vs. temporal sampling as discussed above. However, we feel safe in asserting that this experiment demonstrates the effect, and thus supports the velocity-tuned filter formulation developed earlier.
In the preceding chapters, we have presented a fair amount of theory, and some experimental results, that suggest that velocity-tuned filters are a reasonable way to approach the spatio-temporal approximation task. At this point, we wish to discuss some issues which are relevant to implementation of this theory in human or other vision systems. In essence, what follows is a thought experiment; we wish to extend the results of the previous chapters by suggesting how spatio-temporal approximation may be done in more general cases. In doing so, it is hoped that the important concepts of this thesis will become reinforced by one last iteration through them.

In the following, we are going to make a slight change in terminology, for better or worse. In Chapter 2, we made a clear distinction between "interpolation" and "approximation"; although the two processes are similar in many respects, there are important theoretical differences. In what follows, however, the difference is less important; the important thing is the concept of reconstructing, in some manner, an image which has been sampled. This process seems to be more clearly described by the phrase "interpolating an image" than by the phrase "approximating an image", which may have other confusing connotations. Therefore, please do not be offended if we interchange the two terms from time to time.
5.1 Reconstruction in Real Time-varying Images

First, we need to consider the nature of motion in real images. Our first observation is that objects can move within an image over a wide range of velocities; in fact there are two degrees of freedom, namely velocity magnitude and direction. The second observation is that object motion in an image can give rise to spatial variations in velocity. A good example is the rotation of an object in the image; even though the rotational velocity may be constant over all time, the velocity of a given point on the object varies in magnitude with the distance of the point from the axis of rotation, and changes direction as the object turns.

Now, when we go to the cinema, and see a movie of some rotating object, our visual systems do a good job of interpolating the movie frames; under most conditions, we perceive a smoothly rotating object. Perhaps this is all too obvious, but the conclusion is important; a vision system should be able to perform reconstruction over a wide range of velocities, and should be able to do so locally within in a image.

5.1.1 Velocity-Tuned Channels

The implications of that assertion are as follows. If we are going to allow a wide range of velocities in the input image, and if we are going to use a velocity-tuned spatio-temporal approximation filter approach, then we are probably going to need more than one filter. In principle, one could use only a single filter whose effective velocity bandwidth was large enough to accommodate the entire range of expected velocities. However, this would generally require such a large bandwidth that the filter would not be useful as a reconstruction filter, except in the case of very high sampling rate; recall that in order to be effective, a reconstruction filter must eliminate the spectral replicates from the Fourier transform of the sampled function, which places limits on the maximum allowable bandwidth of the filter.

As suggested by Burr [61], Fahle and Poggio [36], and Poggio et al. [37], a vision system may require a set of velocity-tuned channels, each tuned for some range of velocities. Fahle and Poggio suggested a dichotomy consisting of stasis channels for the analysis of stationary objects, and motion
channels, for the analysis of moving objects; this suggestion is based in part on known psychophysical and neurophysiological properties of biological visual systems. We will not make this distinction here; it would seem that both stasis and motion channels could be implemented with velocity-tuned filters, where in the particular case of a stasis channel, the filter is tuned to velocity $v_f = 0$.

5.1.2 Spatially Local Spatio-temporal Approximation

We will come back to this issue in a moment, but for now, let us consider the issue of spatial variations in velocity. In principle, we could go back to the frequency domain, and try to extend the constant-velocity theorems to handle this situation; it seems likely that one could find special cases where closed-form analytical results could be obtained, especially for such highly structured inputs as constant-velocity rotation or constant-velocity expansion and contraction. Instead, we will argue that we already know enough to deal with this issue.

In Chapter 3, we described an implementation of the spatio-temporal approximation filter in the space-time domain. The motivations for that particular implementation were primarily considerations of computational speed and complexity, but the insights afforded by that implementation are equally valuable. In particular, we recall from that the discussion that the space-time domain implementation could be distilled down to a very local computation; each point in the output was computed as a weighted sum of the points in the input image which were located on a constant-velocity line passing through the corresponding point in the input image. Thus, the algorithm is local in two senses. One is that a relatively small number of points in the input image were used in the computation of each output point; the other is that, in principle, a different constant-velocity line could be used in the computation of each point in the output image. In other words, if we know the velocity at a given point in the input image, we can execute the spatio-temporal approximation algorithm of Chapter 3 using that velocity value; at another point in the image, where the velocity is different, we execute the algorithm using the new velocity value. In principle, we can use a different velocity value at every point in the output image. We note that the local nature of this computation makes its attractive in a parallel implementation such as a biological vision system, or a parallel computer architecture.
5.1.3 Velocity Measurement

There is a major issue hiding in these suggested implementations, which we now feel compelled to expose. To make such an implementation work, one must have a way to decide which velocity-tuned filter to use at a given point in the input image. Alternatively, one could imagine using several velocity-tuned filters in parallel, and selecting the output of the filter which is tuned to the correct velocity. The problem is that in order to do either of these things, one has to know the velocities in the input image, as a function of both position and time. Since the velocity is rarely available as a priori information, in practice, one has to make a velocity measurement. Unfortunately, the best way to measure motion in an image is still unclear, although the subject is a focus of current research; see for example [62].

To be fair, the velocity measurement may not need to be very good. In both the theoretical development and the demonstration of velocity-tuned filters, we have tried to emphasize the concept of effective velocity bandwidth of a filter; we recall that constant-velocity functions can passed nearly unchanged by a velocity-tuned filter if the velocity of the input is within the effective velocity bandwidth of the filter. Thus, as suggested by Fahle and Poggio [36], rather than having a large number of sharply tuned velocity-tuned filters in a vision system, a smaller number of more broadly tuned filters may be sufficient.

Assuming that a good velocity measurement algorithm is available, these suggestions would seem straightforward to implement in a computer-based vision system; with the image stored in a three-dimensional array someplace, we could make the velocity measurements, then perform the spatio-temporal approximation in a “second pass” through the data. Things become harder if the spatio-temporal approximation has to be done in real-time; this statement applies equally well to the case of a biological implementation. However, as we will discuss momentarily, an interesting possibility is that measurement and reconstruction may happen at the same time. To be more precise, the operators which are involved in computing motion in images may also be responsible for spatio-temporal approximation of the image.
5.1.4 Motion Discontinuity

Another reality which must be faced is that moving objects may change direction or speed very quickly; a good example of this is the velocity of a bouncing ball. If we were to implement a general version of the spatio-temporal approximation algorithm as discussed, we might expect some problems with this type of input. After all, the underlying assumption of the spatio-temporal approximation approach is that velocities are approximately constant over small intervals of time and space. This approximation leads to a result which says that it is reasonable to smear the input image in the direction that you think the image is moving. However, this assumption breaks down when objects suddenly change velocity. As a result we might expect that spatio-temporal approximation algorithm to smear the image significantly at velocity discontinuities; indeed, this approach would tend to smooth out changes in velocity.

This issue may not be very significant; after all, for any image processing system, one will be able to derive input images which the system is unable to handle well. However, it has been argued that motion discontinuities may be very valuable primitives for the representation of motion [63]; in cases such as this, when it is important to handle motion discontinuity, there is an approach that may be fruitful.

We refer to the derivation of the regularized solution to spatio-temporal approximation, as discussed in Section 2.4. In that derivation, we are able to control the smoothness of the approximation spline by controlling the order of the derivatives which appear in the regularizing functional. As shown in Figure 2.4.1, as the order increases, the approximation spline becomes smoother, and the number of continuous derivatives of the spline increases. Thus, there is a trade-off in the selection of the regularizing functional. In the context of the edge-detection problem, for example, the filter order should be high enough that we can safely take the necessary derivatives for edge-detection. However, real images often contain sharp intensity discontinuities that are “real” in the sense that they arise from edges of objects or shadows, etc.; unfortunately, the regularized filter smoothes out these discontinuities as well, producing errors in their detection and localization. In essence, the tradeoff is between the need to reduce the noise in the image, which can give rise to false or distorted edges, and the smoothing effect.
of the filter which tends to smear the real edges in the image, usually resulting in errors in localization. Again, the reason for this apparent failure is that the smoothness criterion which was implicit in the choice of the the regularization functional is often invalid in some regions of a particular image.

A recently developed approach to handling this problem is the use of controlled-continuity filters [57]. These filters are based on the regularization approach, but there is a twist. In essence, the algorithm can vary the order of the regularization functional as it deems appropriate, as the filtering operation proceeds. Such an algorithm must have information about the location of discontinuities, either as a priori information, or by measurement; in the region of a discontinuity, a lower order regularizing functional is used, to reduce the smearing effect of the filter.

Clearly, this concept has application to the problem of motion discontinuities. Indeed, we saw that the one-dimensional regularized spatio-temporal approximation filter which came out of the derivation of Section 2.4 was the same as the filter which has been derived elsewhere for the edge-detection problem. Our suggestion, then, is that the problem of motion discontinuity may respond well to the application of controlled continuity filters. This would fit nicely within the framework suggested above. In computing each point in the output image, one would have the choice not only of the velocity of the filter, but also the smoothness. We saw that the order of the regularizing functional used in filter derivation appeared in the result as the order of an auto-correlated Butterworth filter; one could precompute an arsenal of such filters from which to choose as the spatio-temporal approximation operator at a given point in the image.

5.1.5 Causality

In general, we have made little distinction between spatial and temporal filters in this thesis; in many cases, there is no need to. However, there is one distinction that is important, and which has been deftly avoided until now. This issue is that in the real world, temporal filters are usually causal; whereas spatial filters are not.
Section 5.1.5

Mathematically, a \textit{causal filter} is one whose impulse response is identically zero for all times $t < 0$. In words, a causal filter is one which responds to a particular input only after the input has occurred. To see that these descriptions are equivalent, consider the case when the input to a filter is an impulse at time $t = t_0$. The output of the filter would be equal the impulse response of the filter, with its origin shifted to $t = t_0$; if the filter is causal, then the output of the filter will be non-zero for times $t \geq t_0$, \textit{i.e.} only after the impulse has occurred. One can see that causality is a reasonable property for a temporal filter; somehow the concept of a filter responding to an input before it occurs is pretty upsetting. By the same token, causality is less of an issue for spatial filters; there is not a clear sense of "before" and "after" as there is in the temporal domain.

In our derivation of spatio-temporal approximation filters, we have not imposed a causality constraint. We explicitly noted, in fact, that the regularized spatio-temporal approximation filter was sort of a non-causal version of the Butterworth filter. In the context of edge-detection, a non-causal filter makes sense; however, in the context of spatio-temporal approximation, where the filter is in a real sense both a spatial and a temporal filter, perhaps causality should be an issue.

Although we chose to use non-causal filters in the implementation presented in Chapter 3, we wish to point out that the algorithm can be easily adapted to use causal filters. In fact, the adaptation is trivial; we do the same computation as before, but use a filter $f[t]$ which is causal. In effect, when computing the value at a point $[x_0, y_0, t_0]$ in the output image, we consider only those points in the input image which lie "in the past", $t < t_0$; of course, those points will all lie on a constant-velocity line as they did in the non-causal case. In other words, in a discrete implementation, the computation of each frame of the output can only make use of the data in the input frames which have already appeared.

With the advent of computer-based image processing, causality of temporal filters became less of an issue; it became possible to store images, and use causal and non-causal temporal filters with equal ease. Thus, we feel no compulsion to pursue this point further. However, it might be noted that biological vision systems would be constrained to use causal temporal filters; for that reason, it would
be interesting to see what would happen to the regularized spatio-temporal approximation filter if a causality constraint was imposed.

5.2 Velocity Measurements

As mentioned, an important issue in a general implementation of spatio-temporal approximation filters is that we need to have at least an estimate of the velocity field in an image, so that we can choose an appropriate spatio-temporal approximation operator to compute each point in the output image. We do not intend to discuss this topic exhaustively, but there are a few points which should be raised.

We consider a computational framework for computing the velocity field, which was presented by Marr and Ullman [64]; see also [65]. Briefly, the computation has two parts; the image is filtered with a smoothing operator, such as Gaussian, and the LaPlacian of the resulting image is computed. Due to linearity, this operation is equivalent to convolving the image with the LaPlacian of the Gaussian, $\nabla^2 G$. Zero-crossings in the resulting image signal spatial changes in intensity in the original image, at a scale which is determined by the size of the smoothing filter. The second part is to compute the temporal derivative of the smoothed and differentiated image, i.e. $\frac{\partial}{\partial t}[\nabla^2 G * I]$, at the zero-crossings. This measurement determines the motion of the zero-crossing to within 180 degrees; further constraints are added to the local measurements, in order to determine the overall velocity field. An important element of the approach is that motion is detected by directionally selective units, i.e. operators that respond only to motion in one direction, but not to motion in the opposite direction. Subsequent work by Hildreth [62] examined the use of smoothness constraints in combining local velocity measurements to establish the overall velocity field.

In the same paper, Marr and Ullman presented evidence that the such a framework is supported by known neurophysiology of the retina and lateral geniculate nuclei. In particular, the X cells [66], or sustained-response cells, seem to perform a computation similar to $\nabla^2 G * I$; furthermore, under some conditions, the output of the Y cells, or transient-response cells, seems to approximate the bandlimited temporal derivative of that signal.
Other methods of measuring the velocity field have been proposed, such as Schunk [67]; as stated, there is continuing discussion of the relative merits and weaknesses of different approaches. However, our interest is in how these concepts relate to the issue of spatio-temporal approximation. The key to this discussion is the concept of implicit reconstruction, as has been mentioned earlier. In particular, there may indeed be no "spatio-temporal approximation module" in the human visual system, but rather the bandlimited properties of the operators which are used to detect and measure motion may perform spatio-temporal approximation implicitly.

This concept is illustrated in Figure 5.2.1. Here, we have sketched the support of an operator which computes a bandlimited temporal derivative of $\nabla^2 G * I$, as described above. The smoothing filter, e.g. a Gaussian filter, is a spatial low-pass filter; applying the second spatial derivative multiplies its spectrum by $\omega^2$, by the differentiation rule for Fourier transforms, so that the overall spectrum has a bandpass support. Differentiation in time multiplies the spectrum by $-j\omega$; thus a bandlimited differentiation gives the overall spectrum the appearance shown. In the same figure, we illustrate that if the preceding operator is followed by a directionally selective unit, the spectrum becomes zero in two of the four quadrants of the frequency plane. This is easy to understand from the constant-velocity theorem; the support of a filter which is tuned to a positive velocity lies in the second and fourth quadrants of the frequency plane; if the velocity is negative, the support lies in the first and third quadrants.

Having laid this groundwork, we come to the point which we wish to make. The result of passing an image through this operator is indistinguishable from passing the same image through a velocity-tuned filter whose support includes the support of the operator, followed by the operator itself; this is illustrated in Figure 5.2.2. Thus, the operator performs an implicit spatio-temporal approximation of the image at the same time as it is performing its normal job. This concept is implied by the computational framework put forth by Fahle and Poggio [36]. Moreover, Fahle and Poggio show that many of the psychophysical results involving spatio-temporal approximation can be explained well by such a formulation. The importance of this concept can probably not be overemphasized; it connects the results of Chapter 2, which were derived under some very restrictive assumptions, to some very powerful ideas, which are already in place, about how biological visual systems analyze and perceive motion in images. In particular, it establishes a possible link between the framework developed in this thesis, and
Figure 5.2.1 Approximate Support of Y-Cell Operator
Figure 5.2.2 Y-Cell Operator Is Implicit Velocity-Tuned Filter
the whole concept of using zero-crossings in bandpass channels as primitives for edge-detection, and ultimately for image understanding.

5.3 Motion Smear

It has been remarked [68] that if we were to take a photograph with the same exposure time as the "integration time constant" used by the human visual system, i.e. about 100 milliseconds, that most objects moving in the field of view of the camera would appear to be blurred. However, most people would not claim that objects moving in their visual field appear to be blurred, so one may rightly wonder what processing occurs in the human visual system to suppress or remove this motion smear from the perceived image.

Of course, a large component of this effect could be ascribed to tracking movements of the eye; as mentioned earlier, if one is able to track a moving object well, then the image of the object would become nearly stationary on the retina. Of course, this would only apply the object being tracked, and to those objects or image features which happened to be moving at or near the same velocity; even the image of stationary objects would then become blurred by the eye motion. Perhaps objects which are not being tracked, and are thus blurred, are ignored due to some attentive viewing process.

However, it is clear that velocity-tuned filtering could account for some of this effect, even in the absence of eye motion. We saw, both theoretically and experimentally, that if the velocity of an object is within the effective velocity bandwidth of a velocity-tuned filter, then the image of the moving object can be passed by the filter with little or no spatial smearing; this was also true in the context of reconstruction of sampled images. Thus, one may speculate that the action of velocity-tuned filters could account for the relative absence of perceived motion smear.

In the framework of Fahle and Poggio discussed above, the image of a moving object would be passed through a number of velocity-tuned channels, each broadly tuned to a range of velocities; the image of the moving object would be passed with little or no motion blur by a channel which was tuned to a velocity near that of the object, and would be blurred in the other channels. One could then
posit the presence of some gating mechanism which causes the relatively blur-free image to be passed to higher visual centers for analysis, *i.e.* which makes one "pay attention" to the least blurred image. As discussed above, in a more general vision system, the velocity-tuned filters would have to operate in a spatially local manner, to accommodate the spatial variations in the velocity field in real images. Thus, in a similar manner, such a gating mechanism should not act globally, to select the output of one velocity-tuned channel for analysis of the entire image; rather, it should act locally, selecting at each point or region in the image the output of the channel which had the smallest blurring effect in that region of the image.

5.4 Applications

Finally, we wish to mention two specific applications for the concepts presented in this thesis, beyond the general application of computer vision.

5.4.1 Edge-detection in Noisy Image Sequences

We predicted in Chapter 4 that velocity-tuned filters could increase the signal-to-noise ratio in noisy image sequences, potentiating an improvement in the detectability and analysis of the images of moving objects; this prediction was supported by our psychophysical experiments. We now suggest that this property makes velocity-tuned filters potentially useful in analysis of more general, time-varying noisy images. A good example comes from the field of nuclear medicine.

The term *nuclear medicine* refers both to a medical imaging modality, and to the medical specialty in which this imaging is performed. Typically, a nuclear medicine image is obtained by injecting a radioactively labeled *physiologic tracer* into a patient; the tracer accumulates in a particular subpopulation of cells, on the basis of some functional attribute of the cells. The spatial distribution of the radioactivity is then imaged by a scintillation camera. This typically consists of one or more sodium-iodide crystals, which transduce gamma radiation to visible-light photons; the photons are detected in turn by an array of photo-multiplier tubes. The advantage of this type of study over other imaging modalities is that
the distribution of radioactive events is an indicator of tissue function, whereas other modalities such as X-ray computed tomography reveal primarily structural information. The disadvantage is that typical studies result in photon-limited images. Low radioactivity levels, and relatively short imaging times conspire to produce a low number of "counts" in each image pixel; typical values range are less than 1024. Because of the statistical nature of radioactive decay, such low count levels are accompanied by significant noise levels.

A important and very common nuclear medicine study is the radionuclide ventriculogram; this is a dynamic study in which labeled red blood cells are used to images the blood volume inside the heart, for the purpose of evaluating chamber size, wall motion, and pumping efficiency of the heart. Data acquisition for this study is synchronized to the EKG, the electrical signal which indicates the electrical state of the heart tissue in general, and the depolarization and repolarization of the heart during each cardiac cycle, in particular. The data from each cardiac cycle is collected into a number of bins which subdivide the cycle; the result is an image sequence, i.e. a set of discrete frames, which depicts an "average cardiac cycle".

In recent years, analysis of radionuclide ventriculograms has become quantitative, with the use of digital image processing. [69-70]. The first step in the analysis is the identification and demarcation of the chambers of the heart in each frame. Attempts have been made to automate this process, but in general, it is performed by humans; usually one or two frames of the RVG are displayed statically, and the user indicates the chamber boundaries with a light-pen or a joy-stick. Due to the noise in each image, locating the edges in a frame can be very difficult; in general, both interobserver and intraobserver variability are significant. Unfortunately, programs which analyze the RVG data can be very sensitive to the location of the boundaries which are specified by the user.

It has been observed however, that the chamber boundaries are more distinct when the RVG image sequence is displayed as a movie, than when the frames are viewed statically. From our theoretical and experimental results, we can hazard a guess as to why this might be. We have already seen that a velocity-tuned filter can improve the detectability of an object which is moving at the same velocity as the filter. But we have argued, hopefully convincingly, that velocity-tuned filters can be implemented
more generally, to handle cases a range of velocities, as well as spatial variation in velocity. By
implication then, the effect described could be attributed to the effect of locally operating velocity-tuned
filters, i.e. spatio-temporal approximation operators, on an time-varying image which depicts the moving
heart. By further implication, a program intended to localize and identify the chamber walls in an RVG
would benefit from the incorporation of the concept of spatio-temporal approximation; in principle,
and increased signal-to-noise ratio could be achieved over other methods.

5.4.2 Reduction of Transmission Bandwidth

A second possible application is the reduction of transmission bandwidth of time-varying images,
such as television signals. In principle, rather than sending raw intensity values, one could trans-
mit a small number of raw images, in combination with velocity information that would allow the
reconstruction of the time-varying image.

One could imagine, for instance, that at the transmitting end, the complete time-varying image
would be acquired, and the velocity field measurements made. Samples of the image, along with the
velocity information would be sent to the receiver, which would perform a spatio-temporal approximation
to reconstruct the original image. A substantial reduction of transmission bandwidth might be achieved,
especially in cases where large parts of the velocity field was zero. Such an implementation might be
particularly applicable to teleconferencing images, which would not be expected to change much, and
which are to be eventually transmitted over low-bandwidth telephone lines.

An implementation of this concept would clearly require a resolution of the issue of how to best
measure the velocity field. We also point out that, in spirit, this application attempts to represent an
image using information about its derivative; as mentioned in Chapter 1, Shannon remarked that such
schemes tend to be require very high precision in the measurement of derivatives in order to obtain
good results. It seems likely, then, that sensitivity to noise in the original image could be an important
issue, inasmuch as it affects the velocity measurements.
5.5 Conclusion

The intent of this thesis was threefold. The first was to fully develop the theoretical framework for velocity-tuned filters and their application to the spatio-temporal interpolation problem. The second was to show that a velocity-tuned filter formulation had important implications concerning the detection of moving objects in the presence of noise. And finally, we were compelled to show, at least on heuristic grounds, that although the theoretical development was based on rather restrictive assumptions, the main concepts could be extended to much more general cases, ones that would be more interesting in a practical sense.

The key concept, if one had to be so identified, would be that the velocity-tuned filter, either in a spatio-temporal reconstruction context or not, can be implemented as a very simple, local operation, which makes it attractive both for biological and parallel-architecture computer-based vision systems. Running a close second would be that this operation may not be performed explicitly in biological vision systems, but implicitly by operators that are trying to process motion in the retinal image; this concept links the theory of velocity-tuned filters to a large body of knowledge about motion detection and analysis.

In summary, we have attempted to establish a computational framework which is mathematically sound, intuitively pleasing, and consistent the well-known properties of the human visual system in the context of reconstruction of temporally sampled images. In doing do, we have developed a strong case for the use of velocity-tuned spatio-temporal approximation in vision.
A.1 One-dimensional Case

The term *impulse* was introduced to describe any unit-area pulse that is indistinguishable from a briefer unit-area pulse. Implicit in this concept is that one is working with some physical system, such as an electrical circuit, that has finite resolving power and that once the pulse duration (temporal, spatial, *etc.*) goes below the limit of resolution of the system, all pulses of the same area have the same effect on the system.

The impulse is represented symbolically by the Dirac delta, \( \delta(x) \). Clearly, the impulse is not a function *per se*. Rather, it is a member of a class of *generalized functions*, or *distributions*. It is defined not by “what it is”, but by “what it does”. In particular, it is defined by what it does inside an integral.

\[
\delta(x) = 0 \text{ for } x \neq 0, \quad \int_{-\infty}^{+\infty} \delta(x) \, dx = 1
\]

It can be shown that any number of unit-area pulse-shaped functions have this property, in the limit as their width goes to zero, with their area held constant; examples include “box functions”, Gaussian pulses, etc.
Appendix A

Impulses

Of particular importance is the sifting property of the impulse, given in one of two ways as:

\[
\int_{-\infty}^{+\infty} f(x) \delta(x - x_0) \, dx = f(x_0)
\]

\[
\int_{-\infty}^{+\infty} f(x - x_0) \delta(x) \, dx = f(-x_0)
\]

By implication of the sifting property, the convolution of any function with an impulse results in a replication of the function at the location of the impulse:

\[
f(x) \ast \delta(x - x_0) \overset{\text{def}}{=} \int_{-\infty}^{+\infty} f(x' - x) \delta(x' - x_0) \, dx'
\]

\[= f(x - x_0)
\]

The equivalence of small Gaussian pulses and square pulses seems plausible. Somewhat less intuitive is the result that the integral of complex exponentials behaves as an impulse:

\[
\int_{-\infty}^{+\infty} e^{iwx} \, dw = \delta(x)
\]

By this is meant only that this function exhibits the properties described above, when placed inside an integral. In addition, by a similarity transform:

\[
\int_{-\infty}^{+\infty} e^{iawx} \, dw = \frac{1}{|a|} \delta(x)
\]

These types of impulses will be important in our study of constant-velocity functions and the Constant-velocity Theorem.
Appendix A

Impulses

Strictly speaking, the area under any impulse is 1. However, if an impulse is multiplied by some coefficient, then the area under the resulting function equals the coefficient; consequently, the coefficient is often referred to as the area of the impulse. This confusing notation will be avoided in this thesis, but we will refer to the unit impulse, which is an impulse whose coefficient is 1.

In the limit of zero width, the height of an impulse gets arbitrarily large, which makes it someone hard to depict graphically. We will use the convention of using an arrow to represent an impulse, and letting the height of the arrow represent the coefficient of the impulse, i.e., the area underneath it.

A.2 Two-dimensional Case

Two-dimensional impulses are a straightforward extension of the one-dimensional impulse, both conceptually and mathematically. Conceptually, a two-dimensional impulse is the limit of any two-dimensional pulse as its width and height go to zero, with constant volume underneath it. The properties of two-dimensional impulses are predictable extensions of the one-dimensional properties above:

\[
\delta(x, y) = 0 \text{ for } x \neq 0 \text{ and } y \neq 0, \quad \iint_{-\infty}^{+\infty} \delta(x, y) \, dx \, dy = 1
\]

The two-dimensional sifting property is also a straightforward extension of the one-dimensional case:

\[
\iint_{-\infty}^{+\infty} f(x, y) \, \delta(x - x_0, y - y_0) \, dx \, dy = f(x_0, y_0)
\]

\[
f(x, y) \ast \delta(x - x_0, y - y_0) \overset{\text{def}}{=} \iint_{-\infty}^{+\infty} f(x', x) \, \delta(x' - x_0, y - y_0) \, dx' \, dy' = f(x - x_0, y - y_0)
\]
As in the one-dimensional case, two-dimensional impulses include some intuitive functions, such as two-dimensional Gaussians and box functions, as well as this less intuitive form involving two-dimensional complex exponentials:

\[
\int_{-\infty}^{+\infty} e^{iawx} e^{i\omega xy} d\omega_x d\omega_y = \frac{1}{|ab|}\delta(x, y)
\]

Clearly, two-dimensional impulses are not interchangeable with one-dimensional impulses; their properties are defined with respect to two and one-dimensional integrals, respectively. Thus, putting a one-dimensional impulse inside a double integral is usually not useful. To distinguish one-dimensional and two-dimensional impulses, we will use the notation: \(\delta_1()\) for the one-dimensional case, and \(\delta_2()\) for the two-dimensional case. Interestingly, it can be shown that two-dimensional impulses can be decomposed into one-dimensional impulses:

\[
\delta_2(x, y) = \delta_1(x)\delta_1(y)
\]

In two dimensions, one can also encounter some other geometries. Of particular interest in this thesis is the impulse line, which has the form of a line of one-dimensional impulses, traversing the two-dimensional plane. For example, the following impulse line:

\[
\delta_1(ax + t)
\]

is non-zero only along the line: \(ax + t = 0\); at each point along the line, we find a one-dimensional impulse. Conceptually, the impulse line is the limit, as width goes to zero, of a ridge-shaped function in two dimensions; that is, whereas \(\delta_2()\) has a pulse-shape in all cross sections, the impulse line, which we will denote by \(\delta_1()\), is pulse-shaped in only one dimension, and extends indefinitely along its axis. Because of this, it is meaningless to talk about the area or volume of the impulse line. In fact, this geometry does not share many similar traits with \(\delta_2()\). We can, however, characterize its behavior by
placing it inside various integrals. Rather than address this issue here, we will consider it in considerable depth when we look at velocity-tuned filters.

In three dimensions, we will see this concept extended, and encounter both impulse lines and impulse plane. For example, the following impulse plane described by:

\[ \delta_p (ax + by + t) \]

is non-zero only along the plane described by: \( ax + by + t = 0 \), which goes through the origin of three-space. At each point along this plane, we find a one-dimensional impulse.

An impulse line in three-space looks a little different from the one we saw before; for example, the impulse line described by:

\[ \delta_l (ax + t, by + t) \]

is non-zero only when both of the arguments are zero. The equations \( ax + t = 0 \) and \( by + t = 0 \) each describe a plane in three-space, and their intersection is a line; along this line we find a two-dimensional impulse.

The properties of these impulse lines and impulse planes are also revealed by putting them inside integrals; we will explore this fully elsewhere, in the context of three-dimensional velocity-tuned filters.
A.3 Discrete Case

In the discrete case, the definition of an impulse is a little more intuitive:

\[
\delta_1[n] = \begin{cases} 
1 & \text{for } n = 0 \\
0 & \text{else.} 
\end{cases}
\]

This function has similar properties to the continuous case, where summation replaces integration. Thus, the sifting property becomes:

\[
\sum_{n=-\infty}^{+\infty} x[n] \delta_1[n - n_0] = x[n_0]
\]

The two-dimensional case is a straightforward extension, although the one-dimensional and two-dimensional discrete impulses are more similar than in the analog case. They only differ in the dimension of the domain in which they are found.

\[
\delta_2[m, n] = \begin{cases} 
1 & \text{if } m = 0 \text{ and } n = 0 \\
0 & \text{else.} 
\end{cases}
\]

\[
\sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} x[m, n] \delta_2[m - m_0, n - n_0] = x[m_0, n_0]
\]

Once again, we can also encounter impulse lines and impulse planes in the discrete domain. These are lines and planes in higher-dimensional discrete spaces, along which one-dimensional discrete impulses are found; they are somewhat easier to visualize than their continuous counterparts. Their properties are essentially the same as in the continuous case, except that they are revealed by multi-dimensional summation signs, rather than integrals.
Appendix B

Sampling Theorems

There are two fundamental questions in sampling theory: (1) What class of functions, if any, can be completely characterized by a set of discrete samples; and (2) how can the original signal be reconstructed from those samples. These questions are addressed by a set of theorems known collectively as sampling theorems. The Classic Sampling Theorem, as stated by Shannon [2], says: "If a function \( f(t) \) contains no frequencies higher than \( W \) cycles per second, it is completely determined by giving its ordinates at a series of points spaced \( 1/(2W) \) apart." We will prove a somewhat more general form of this theorem, in a way that is fairly enlightening, inasmuch as it tells us how to reconstruct the original signal, and which introduces concepts that we will draw upon throughout the thesis.

B.1 One-dimensional Functions

Let \( x(t) \) represent some continuous function, and let \( x_s(t) \) be a continuous signal formed by multiplying the function \( x(t) \) by an impulse train. An impulse train is just a set of impulses, located at various times, thus:

\[
\sum_{n=-\infty}^{+\infty} \delta(t - \tau_n)
\]
Appendix B

Sampling Theorems

The product will be another impulse train, and the coefficient of each impulse will equal the value of \( x(t) \) at the corresponding sample time. Let \( x[n] \) be the coefficient of the impulse at \( t = \tau_n \).

\[
x_p(t) = x(t) \times \sum_{n=-\infty}^{+\infty} \delta(t - \tau_n)
\]

\[
= \sum_{n=-\infty}^{+\infty} x(nT) \delta(t - \tau_n) \overset{\text{def}}{=} \sum_{n=-\infty}^{+\infty} x[n] \delta(t - \tau_n)
\]

Now, assume that the Fourier transform of \( x(t) \) exists, (which restricts the class of signals for which this particular Sampling Theorem applies), and is given by :

\[
X(f) = \int x(t)e^{-j2\pi ft} dt
\]

It can be shown that the Fourier transform of the product of two functions is given by the convolution of the Fourier transforms of those functions :

if : \( s(t) = x(t) h(t) \)

then : \( S(f) = X(f) * H(f) \overset{\text{def}}{=} \int_{-\infty}^{+\infty} H(f') g(f - f') df' \)

Since we have expressed \( x_p(t) \) as a product, the Fourier transform of the sampled function can be found by convolution in this manner; in this case, the Fourier transform of the original signal is convolved by the Fourier transform of the impulse train.

Now, in the general case, the Fourier transform of the impulse train can be rather hard to evaluate. However, we most commonly use periodic sampling, by which we mean that the unit impulses occur periodically with period \( T \), so that \( \tau_n = nT \). The Fourier transform of this function is rather easy to find; it is a periodic impulse train whose period equals the sampling frequency, \( F_s = 1/T \).
Now impulses, as described in Appendix A, have the property that when convolved with any other function, that function is replicated at the location of the impulse. Thus, when we convolve the original spectrum $X(f)$ with this impulse train, the effect is to replicate this spectrum at frequency intervals of $1/T$:

$$X_s(f) = X(f) \ast \sum_{n=-\infty}^{+\infty} \delta\left(f - \frac{n}{T}\right)$$

$$= \sum_{n=-\infty}^{+\infty} X\left(f - \frac{n}{T}\right) \delta\left(f - \frac{n}{T}\right)$$

This is illustrated in Figure B.1.1. Because of the uniqueness property of Fourier transforms, the original function will be recoverable if and only if we can recover the spectrum of the original signal, i.e. the $n = 0$ term in the expression above, from the rest of terms, which are the spectral replicates. In general, these replicates will mix together, making the original spectrum unrecoverable. However, there is an exception: If the original signal becomes identically zero for frequencies above some frequency $W$, i.e. if the signal is bandlimited to $W$, and if the spectral replicates are far enough apart, then there will be no overlap, and the original spectrum will be, in theory, recoverable. Clearly, the spacing between the replicates must be greater than or equal to the total extent of the non-zero portion of the original spectrum. From this observation, we derive a necessary and sufficient condition on the sample frequency: $1/T \geq 2W$. Inverting produces an equivalent condition on the sampling interval: $T \leq 1/(2W)$.

Of course, the Classic Sampling Theorem describes a special case where $T = 1/2W$; the above arguments outline a more rigorous proof of the Classic Sampling Theorem.

However, there is more to be revealed. In particular, we can deduce a method of recovering the original signal. Suppose we multiply the spectrum of the sampled signal by a function that has unity value over the support of the original spectrum, i.e. the frequencies for which the spectrum is non-zero, and zero elsewhere. Due to the uniqueness of the Fourier transform, if we find the inverse Fourier transform of this new function, we will recover the original signal exactly.
Figure B.1.1 Sampling as Viewed in the Frequency Domain
The mathematical operation just described can be performed, in theory, by a linear filter. A linear filter can be characterized by a function called its system function; the Fourier transform of the output of a linear filter equals the product of the system function and the Fourier transform of the input. The inverse Fourier transform of the system function is called the impulse response of the filter; this is the output of the filter when the input is an impulse. Now, it can be shown that the inverse Fourier transform of the product of two frequency-domain functions is the convolution of each of the inverse Fourier transforms of each of the individual functions. Thus, the output of a linear filter is also equal to the convolution of the input function with the impulse response of the filter.

Now, to recover the spectrum of our original signal, we can use an ideal low-pass linear filter, which has a system function, \( H(f) \), as shown below; the impulse response \( h(t) \) of this filter has the form of a sinc, or \((\sin x)/x\) function; such a filter is illustrated in Figure B.1.2.

\[
H(f) = \begin{cases} 
1, & \text{if } |f| < W \\
0, & \text{otherwise.}
\end{cases} \quad \Rightarrow \quad h(t) = 2W \frac{\sin 2\pi W t}{2\pi W t}
\]

Passing the sampled function through this filter has the effect of convolving \( x_s(t) \) and \( h(t) \). We recall that \( x_s(t) \) has the form of an impulse train; thus the effect of the convolution is to replicate the sinc functions at the points where each impulse is found.

\[
x(t) = \int_{-\infty}^{+\infty} x_s(t) \ast h(t - \tau)
\]

\[
= \sum_{n=-\infty}^{+\infty} x[n] \delta(t - nT) \ast h(t)
\]

\[
= \sum_{n=-\infty}^{+\infty} x[n] 2W \frac{\sin 2\pi W (t - nT)}{2\pi W (t - nT)}
\]

We conclude that a bandlimited \( x(t) \) can be reconstructed from its samples by adding together a number of sinc functions, scaled by the sample values, and shifted by the sample delays. An equivalent
Figure B.1.2 Reconstruction of Original Function From Its Samples
viewpoint is that we reconstruct \( x(t) \) by interpolating to find the values between the sample points in \( x_s(t) \); we do this by computing a weighted average of continuous interpolating functions, which in this case are the sinc functions.

As we discovered, the sampling rate need not equal \( 2W \) (known as the Nyquist Rate), but can be anything greater than this value. In addition, if the sampling rate exceeds \( 2W \), then the cutoff frequency of the low-pass filter need not be exactly \( W \), but can be anywhere between the high end of the original spectrum, and the low end of its first replicate, i.e. between \( W \) and \((F_s - W)\). Perhaps more importantly, in this case, an ideal low-pass filter is not needed for reconstruction; rather any function that is flat up to frequency \( W \), and goes to zero in some fashion in the range \((F_s - W)\) is sufficient. Because ideal low-pass filters are impossible to implement, and even hard to approximate, sampling above the Nyquist rate is clearly desirable. However, the trade-off is that faster sampling rates produce larger number of samples, and these excess samples are redundant, i.e. they contain no new information about the original signal. However, in many instances, notably noisy ones, redundancy is desirable as way of assuring robustness.

### B.2 Aliasing

We saw that the ability to reconstruct our original function depended on having non-overlapping spectral replicates in the sampled function. This condition will not be met if the sampling rate falls below the Nyquist rate, or equivalently, if the signal is not sufficiently bandlimited for a given sampling rate. As the sampling rate falls, the spectral replicate get closer and closer; at rates below the Nyquist rate, the spectra begin to overlap and are hopelessly mixed together. When the reconstruction filter is applied, the result will contain not only the original spectrum, but pieces of neighboring spectral replicates mixed in. Consequently, the original signal will not be recovered.

This phenomenon is referred to as aliasing, and is illustrated in Figure B.2.1. This is presumably because the spectrum of the undersampled signal is indistinguishable from that of a bandlimited signal sampled at the Nyquist Rate; thus the corresponding time-domain functions are really the same function.
Figure B.2.1 Aliasing and Partial Recovery of Undersampled Functions
but derived in different ways. Indeed, there would be an infinitely large set of undersampled functions having the same spectrum, and thus being "aliases" of the same function.

Because each replicate is shifted in frequency relative to the original spectrum, mixed-in replicates will appear at different frequency than they were originally. This often leads to the appearance of frequency components in the undersampled function that were not in the original one. For instance a 10 Hz frequency component in the original signal may be shifted to 1 Hz in the replicate, and thus be added to the original signal as a 1 Hz component.

One approach to avoiding aliasing is to pre-filter the continuous signal before sampling to be sure that it is sufficiently bandlimited for the sampling rate. From the arguments above, one can see that the required bandwidth of such a pre-filter is $W \leq 1/2T$. This can also be seen by inverting the Nyquist rate: $1/T \geq 2W$. Further information about optimal pre-filters abounds in the literature; see [6] and [71] for example.

From an information theoretical view, we have determined that $2JW$ samples are needed to capture the information in the original function. By sampling below the Nyquist Rate, we collect less numbers than are necessary, so we lose some of the information in the function; clearly once this has occurred, no amount of data manipulation can recover the lost information.

It should be pointed out that, although aliasing does irreversible damage, we may be able to recover part of the original signal. In particular, if sampling rate is greater than one-half the Nyquist Rate, then there will be some part of the original spectrum left intact; there will be no replicates mixed in over these frequencies. Thus, if we use a filter whose bandwidth is $(1/T - W)$, we are guaranteed to eliminate all spectral replicates, and pass only the uncontaminated portion of the original spectrum; the result will be the same as if we had passed the original signal through the same low-pass filter. The topic of optimum post-filtering is also discussed in the literature; see [6] for example.
B.3 Implicit Reconstruction

An important observation is that if a sampled function is passed through any bandlimited filter, i.e. not necessarily an ideal low-pass filter, whose bandwidth is small enough to eliminate the spectral replicates in the Fourier transform of the sampled function, then the result is the same as passing the original function through the same filter. In this case, no explicit reconstruction step is needed; reconstruction happens 'automatically' by virtue of the bandlimitedness of this operator. In fact, since ideal reconstruction filters cannot be realized, there is good reason to intentionally omit an explicit reconstruction step, inasmuch as one would be forced to use a filter which only approximated the ideal reconstruction filter, thus unavoidably introducing some kind of errors.

B.4 Two-dimensional Sampling and Reconstruction

These results can be extended to two or more dimensions in a straightforward manner. A two-dimensional function \( x(s, t) \) can be sampled in either \( s \) or \( t \) or both.

In the one-dimensional case, we started by multiplying our signal by an impulse train; in the two-dimensional case, this becomes a two-dimensional grid of impulses. This added dimension introduces some sampling geometries that are not available in the one-dimensional case. As before, multiplication by this impulse grid is equivalent to convolution of the original spectrum by the Fourier transform of the grid.

When the impulse grid is periodic, the transform of the impulse grid is also a periodic impulse grid, so this convolution results in the replication of the (two-dimensional) Fourier transform of the original signal along the axis (or axes) in the \((f_s, f_t)\) plane corresponding to the variable in which the function is sampled. As before, we can use geometric arguments to determine the conditions under which reconstruction of the signal is theoretically possible.

If a function is bandlimited to some region such that \( X(f_s, f_t) = 0 \) for \(|f_s| > W_s\) , and if we sample the signal in \( s \) such that the sampling frequency satisfies : \(1/T_s \geq 2W_s\), then there should be
no overlap between spectral replicates, and the original signal will be recoverable; a similar statement holds for the $t$ dimension. When the original signal is bandlimited in both $s$ and $t$, then we can sample in both variables, and as long we sample at or above the Nyquist Rate for both variables, we can theoretically recover the original signal.

For example, a common sampling technique is rectangular sampling, so called because the impulses are found at the center of rectangles in the $(s,t)$ plane. The Fourier transform of the rectangular impulse grid is also a rectangular impulse grid, so that when a signal is sampled in this manner, its original spectrum is replicated on rectangular grid in the $(f_s, f_t)$ plane. The rectangular sampling grid and its Fourier transform are described by:

\[
s_r(s,t) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(s - mT_s, t - nT_t)
\]

\[
S_r(f_s, f_t) = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} \delta(f_s - \frac{m}{T_s}, f_t - \frac{n}{T_t})
\]

This is illustrated in Figure B.4.1. By analogy to the one-dimensional case, we see that if the original signal is bandlimited to: $|f_s| > W_s$ and $|f_t| > W_t$, then there will be no overlap of the spectral replicates, as long as the sampling rates satisfy: $1/T_s \geq 2W_s$ and $1/T_t \geq 2W_t$. If this is the case, then we can theoretically recover the original spectrum from its replicates, by using a (two-dimensional) low-pass filter, which has a transfer function of the form:

\[
H(f_s, f_t) = \begin{cases} 
1, & \text{if } |f_s| < W_s \text{ and } |f_t| < W_t \\
0, & \text{otherwise}.
\end{cases}
\]

The two-dimensional impulse response of this filter is found as the two-dimensional inverse Fourier transform of this system function, and it has the form of the product of two sinc functions. In like manner as the one-dimensional case, applying this filter results in the two-dimensional convolution of the in sampled input function with this impulse response, and again produces a weighted sum of time-shifted sinc functions:
Figure B.4.1 Rectangular Sampling Grid
\[ x(s, t) = \sum_{m = -\infty}^{+\infty} \sum_{n = -\infty}^{+\infty} x[m, n] \frac{2\pi W_s(s - mT)}{2\pi W_s(s - mT)} \frac{\sin 2\pi W_t(t - nT)}{2\pi W_t(t - nT)} \]

The previous statements about reconstruction filters apply. Ideal low-pass filters can be implemented only approximately; as the sampling rates increases above their respective Nyquist Rate, the requirements imposed on the reconstruction filter are relaxed, and robustness in the face of noise is traded for decreased sampling efficiency.

This rectangular sampling scheme is certainly sufficient for most applications, however other schemes are possible. The motivation for using a different sampling pattern is that rectangular sampling is inefficient if the original spectrum does not have a rectangular-shaped support; inefficiency in this case means that we must take a larger number of samples than is really necessary in order to capture the information in the signal. As in the one-dimensional case, as the sampling rate decreases, the spectral replicates get closer together; conversely, the minimum sampling rate is obtained by making the replicates as close together as possible, as long as they do not overlap. Thus, for example, if the support of the original spectrum is a regular hexagon, the most efficient sampling will replicate the original spectrum on a regular hexagonal grid, and in such a way that they “almost touch”. Any other sampling grid will place the spectral replicates farther apart, and will thus require sampling at a higher, and thus higher than necessary, rate.

In the general case shown, the hexagonal grid and its Fourier transform are given by:

\[ s_h(s, t) = \sum_{m = -\infty}^{+\infty} \sum_{n = -\infty}^{+\infty} \delta(s - (m - \frac{n}{2})T_s, \ t - nT_t) \]

\[ S_h(f_s, f_t) = \sum_{m = -\infty}^{+\infty} \sum_{n = -\infty}^{+\infty} \delta(f_s - \frac{m}{T_s}, f_t - (m - \frac{n}{2})\frac{1}{T_t}) \]

The impulses occur at the centers of hexagons in the \((s, t)\) and \((f_s, f_t)\) planes, respectively. In the special case of regular hexagonal sampling, these hexagons are regular, i.e. all sides and included angles
are equal; in this case, each impulse is equidistant from its six nearest neighbors, at some distance $T_h$. One can show that the Fourier transform of this grid is also a regular hexagonal impulse grid, where the nearest-neighbor distribution is $1/T_h$.

A common application of regular hexagonal sampling is the case when the function to be sampled is bandlimited to a circular region of the $(f_x, f_y)$ plane. In this case, hexagonal sampling is more efficient than rectangular sampling because the spectral replicates can be packed closer together without touching on a hexagonal grid. This is illustrated in Figure B.4.2. From the above result, we see that if the circular support has diameter $D$, the most efficient sampling occurs when the distance between neighboring samples is $1/D$; with this sampling interval, the replicates almost touch, so this is the equivalent Nyquist rate for this geometry. A good example of this scenario occurs in vision; the optics of the eye ensure that the retinal image is approximately circularly bandlimited, and the retinal receptors in the fovea are arranged in an approximately regular hexagonal pattern.
Figure B.4.2 Hexagonal Sampling Grid
Appendix C

Constant-velocity Theorems

Theorem C.1 Two-dimensional Continuous Case

If a two-dimensional function, \( g(x, t) \) can be expressed as a constant-velocity translation of a one-dimensional function \( f(x) \), so that \( g(x, t) = f(x - vt) \), then the Fourier transform of \( g(x, t) \) is given by:

\[
G(\omega_x, \omega_t) = F(\omega_x) \delta_1(\omega_x v + \omega_t) \\
= F\left(\frac{-\omega_t}{v}\right) \delta_1(\omega_x v + \omega_t)
\]

Remarks: \( G(\omega_x, \omega_t) \) has the form of an impulse line, as described in Appendix A. It is a line formed by moving a one-dimensional impulse across the \((\omega_x, \omega_t)\) plane at constant-velocity. At each point in the plane were the argument is zero, i.e. when \( \omega_x v + \omega_t = 0 \), we find a one-dimensional impulse, whose coefficient is given by the value of \( F(\omega_x) \), i.e. the Fourier transform of the zero-velocity one-dimensional function, at that point. When \( v = 0 \), the support of \( G(\omega_x, \omega_t) \) is colinear with the \( \omega_x \)-axis. As \( v \) increases, we can imagine the support "rotating" off the axis; it still goes through the origin, but it now has slope \(-v\). However, in this framework, the transformation is not purely rotational. Since the coefficient of each impulse is always \( F(\omega_x) \), independent of velocity, this transformation is such that the projection of \( G(\omega_x, \omega_t) \) onto the \( \omega_x \)-axis remains constant; in fact, it equals \( F(\omega_x) \).
Proof: The two-dimensional Fourier transform of $g(x,t)$ is:

$$G(\omega_x, \omega_t) \overset{\text{def}}{=} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x,t) e^{-j\omega_x x} e^{-j\omega_t t} \, dx \, dt \tag{C.1.a}$$

$$= \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x-vt) e^{-j\omega_x x} \, dx \right] e^{-j\omega_t t} \, dt \tag{C.1.b}$$

Now, we make the substitution: $x' = x - vt$. Since $v$ and $t$ are constants within the first integral, $dx' = dx$. So we have:

$$G(\omega_x, \omega_t) = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x') e^{-j\omega_x (x'+vt)} \, dx' \right] e^{-j\omega_t t} \, dt \tag{C.1.c}$$

$$= \int_{-\infty}^{+\infty} f(x') e^{-j\omega_x x'} \, dx' \int_{-\infty}^{+\infty} e^{-j(\omega_x v + \omega_t) t} \, dt \tag{C.1.d}$$

Now, we recognize the first integral as the Fourier transform of $f(x)$. The second integral behaves as an impulse line, as discussed in Appendix A. So, we may write:

$$G(\omega_x, \omega_t) = F(\omega_x) \delta_t (\omega_x v + \omega_t) \tag{QED}$$

We can also get this result by invoking the shifting theorem for Fourier transforms after step C.1.b.

To get the second form of the result, we rewrite C.1.b, this time integrating first with respect to $t$. Again, let: $x' = x - vt$; then with $v$ and $x$ constant, $dx' = -v \, dt$. The limits of the integral will also switch, depending on the sign of $v$ in such a way that is best summarized by putting absolute value bars around $v$ as shown.
\[
G(\omega_z, \omega_t) = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x') e^{-j \omega_t \left( \frac{x'-z}{v} \right)} \frac{dx'}{|v|} \right] e^{-j \omega_z z} \, dx 
\]
\[= \frac{1}{|v|} \int_{-\infty}^{+\infty} f(x') e^{-j \left( \frac{\omega_t}{v} \right) x'} \, dx' \int_{-\infty}^{+\infty} e^{-j \left( \frac{\omega_z}{z} + \omega_t \right) z} \, dx \]  \hspace{1cm} C.1.e

The first integral is the Fourier transform of \( f(t) \), with \( \omega_t \) replaced by \(- \omega_t/v\). The second integral is very similar to the impulse line in the previous derivation. However, it can be shown that they are not identical; rather a similarity transform is needed. We let \( x'' = x/v \), and \( dx'' = dx/v \), and use absolute value bars to handle the possible switch of limits due to the sign of \( v \) :

\[
G(\omega_z, \omega_t) = \frac{1}{|v|} \int_{-\infty}^{+\infty} F\left( -\frac{\omega_t}{v} \right) e^{-j \left( \omega_t + \omega_z v \right) x''} \frac{dx''}{|v|} \]
\[= F\left( -\frac{\omega_t}{v} \right) \delta_t (\omega_z v + \omega_t) \]  \hspace{1cm} C.1.f

This result may seem obvious. After all, as a result of the impulse line, \( G(\omega_z, \omega_t) \) is non-zero only when \( \omega_t = -\omega_z v \); making this substitution in the first form of the Constant-velocity Theorem gives the second form. It is comforting that both approaches yield the same result.

**Theorem C.2** Three-dimensional Continuous Case

If a three-dimensional function, \( g(x, y, t) \) can be expressed as a constant-velocity translation of a two-dimensional function, \( f(x, y) \), so that \( g(x, y, t) = f(x - v_xt, y - v_y t) \), then the Fourier transform of \( g(x, y, t) \) is given by :

\[
G(\omega_z, \omega_y, \omega_t) = F(\omega_z, \omega_y) \delta_p (\omega_z v_z + \omega_y v_y + \omega_t) 
\]
**Remarks**: $G(\omega_x, \omega_y, \omega_t)$ has the form of an impulse plane, as described in Appendix A. It is zero everywhere in $(\omega_x, \omega_y, \omega_t)$ space, except on the plane described by: $\omega_x v_x + \omega_y v_y + \omega_t = 0$. At each point on this plane, we find a one-dimensional impulse, whose coefficient is given by the value of $F(\omega_x, \omega_y)$, i.e., the Fourier transform of the two-dimensional, zero-velocity function, at that point.

Similarly to the two-dimensional case, we may picture $G(\omega_x, \omega_y, \omega_t)$ as a rotation of the spectrum of the zero-velocity function off of the $(\omega_x, \omega_y)$ plane; it still goes through the origin, but its slope is governed by the $x$ and $y$ components of the velocity. Again, this transformation is such that the projection of $G(\omega_x, \omega_y, \omega_t)$ onto the $(\omega_x, \omega_y)$ plane is always constant, and equals $F(\omega_x, \omega_y)$.

**Proof**: The derivation of a Constant-velocity Theorem in for two-dimensional functions is a straightforward extension of the one-dimensional case. The three-dimensional Fourier transform of $g(x, y, t)$ is:

$$G(\omega_x, \omega_y, \omega_t) \overset{\text{def}}{=} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y, t) e^{-j\omega_x x} e^{-j\omega_y y} e^{-j\omega_t t} \, dx \, dy \, dt \quad \text{C.2.a}$$

$$= \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x - v_x t, y - v_y t) e^{-j\omega_x x} e^{-j\omega_y y} \, dx \, dy \right] e^{-j\omega_t t} \, dt \quad \text{C.2.b}$$

Making substitutions as before: $x' = x - v_x t$, and $y' = y - v_y t$; with $t$ constant inside the first two integrals, $dx' = dx$, and $dy' = dy$. Then we get:

$$G(\omega_x, \omega_y, \omega_t) = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x', y') e^{-j\omega_x (x' + v_x t)} e^{-j\omega_y (y' + v_y t)} \, dx' \, dy' \right] e^{-j\omega_t t} \, dt \quad \text{C.2.c}$$

$$= \int_{-\infty}^{+\infty} f(x', y') e^{-j\omega_x x'} e^{-j\omega_y y'} \, dx' \, dy' \int_{-\infty}^{+\infty} e^{-j(\omega_t + \omega_x v_x + \omega_y v_y)t} \, dt \quad \text{C.2.d}$$

215
Again, the first integral is the two-dimensional Fourier transform of \( f(x, y) \), and the second integral is an impulse line in \((\omega_x, \omega_y, \omega_t)\) space. Thus:

\[
G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) \delta_t (\omega_x v_x + \omega_y v_y + \omega_t) \tag{QED}
\]

**Theorem C.3** Duality for Two-dimensional Continuous Case

If a two-dimensional function, \( g(x, t) \) has a Fourier transform which has the form of a constant-velocity translation of a one-dimensional function, \( F(\omega_t) \), so that \( G(\omega_x, \omega_t) = F(\omega_x v + \omega_t) \), then the function \( g(x, t) \) is given by:

\[
g(x, t) = f(t) \delta_t (x - vt) \\
= f\left(\frac{x}{v}\right) \delta_t (x - vt)
\]

**Remarks**: In like manner as the first theorem, \( g(x, t) \) has the form of a line of one-dimensional impulses in the \((x, t)\) plane. The coefficient of each of the impulses is given by the value of \( f(t) \), i.e., the inverse Fourier transform of the zero-velocity frequency-domain function, at that point.

**Proof**: We start with the definition of the two-dimensional inverse Fourier transform of \( G(\omega_x, \omega_t) \), and make the substitutions: \( \omega'_t = \omega_x v + \omega_t \). Integrating first with respect to \( \omega_t \), we have \( \omega_x \) constant, so \( d\omega'_t = d\omega_t \). Thus:

\[
g(x, t) \overset{\text{def}}{=} \iiint_{-\infty}^{+\infty} G(\omega_x, \omega_t) e^{+j\omega_x t} e^{+j\omega'_t} d\omega_x d\omega_t \tag{C.3.a}
\]

\[
= \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} F(\omega_x v + \omega_t) e^{+j\omega'_t} d\omega_t \right] e^{+j\omega_x t} d\omega_x \tag{C.3.b}
\]

\[
= \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} F(\omega'_t) e^{+j(\omega'_t - \omega_x v)t} d\omega'_t \right] e^{+j\omega_x t} d\omega_x \tag{C.3.c}
\]
Appendix C

Constant-velocity Theorems

\[ \int_{-\infty}^{+\infty} F(\omega_t) e^{+j\omega_t t} \, d\omega_t \int_{-\infty}^{+\infty} e^{+j\omega_x(x-vt)} \, d\omega_x \quad C.3.d \]

We recognize the first integral as the inverse Fourier transform of \( F(\omega_x) \). The second integral is an impulse line, this time in \((x,t)\) space. So, we may write:

\[ g(x,t) = f(t) \delta_t (x-vt) \quad QED \]

To get the second form, we note that the impulse function is zero everywhere except for \( x = vt \), so we conclude that:

\[ g(x,t) = f\left(\frac{x}{v}\right) \delta_t (x-vt) \quad QED \]

A more rigorous proof would proceed as in the first theorem.

**Theorem C.4** Duality for Three-dimensional Continuous Case

If a three-dimensional function, \( g(x,y,t) \) has a Fourier transform which has the form of a constant-velocity translation of a one-dimensional function, \( F(\omega_t) \), so that \( G(\omega_z, \omega_y, \omega_t) = F(\omega_z v_z + \omega_y v_y + \omega_t) \), then the function \( g(x,y,t) \) is given by:

\[ g(x,y,t) = f(t) \delta_t (x + v_z t, y + v_y t) \]

**Remarks**: \( g(x,y,t) \) has the form of a line of two-dimensional impulses in three-dimensional \((x,y,t)\) space; this is in contrast to previous cases, where we had a plane of one-dimensional impulses. The coefficient of each impulse is given by the value of \( f(t) \), i.e. the inverse Fourier transform of the zero-velocity function, at that point.

217
Proof: The two-dimensional inverse Fourier transform of $G(\omega_x, \omega_y, \omega_t)$ is:

$$g(x, y, t) \overset{\text{def}}{=} \iint_{-\infty}^{+\infty} G(\omega_x, \omega_y, \omega_t) e^{+j\omega_x x} e^{+j\omega_y y} e^{+j\omega_t t} d\omega_x d\omega_y d\omega_t \quad C.4.a$$

$$= \iint_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} F(\omega_x v_x + \omega_y v_y + \omega_t) e^{+j\omega_t t} d\omega_t \right] e^{+j\omega_x x} e^{+j\omega_y y} d\omega_x d\omega_y \quad C.4.b$$

We integrate first with respect to $\omega_t$, making the substitution: $\omega_t' = \omega_x v_x + \omega_y v_y + \omega_t$. Inside the first integral, $\omega_x$ and $\omega_y$ are constant, so $d\omega_t' = d\omega_t$:

$$g(x, y, t) = \iint_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} F(\omega_t') e^{+j(\omega_t'-\omega_x v_x - \omega_y v_y)t} d\omega_t' \right] e^{+j\omega_x x} e^{+j\omega_y y} d\omega_x d\omega_y \quad C.4.c$$

$$= \int_{-\infty}^{+\infty} F(\omega_t') e^{+j\omega_t't} d\omega_t' \int_{-\infty}^{+\infty} e^{+j\omega_x (x-v_xt)} e^{+j\omega_y (y-v_yt)} d\omega_x d\omega_y \quad C.4.d$$

The first integral is recognized as an inverse Fourier transform. The second represents a line of two-dimensional impulses, moving across the $(x, y)$ plane at constant velocity. Thus:

$$g(x, y, t) = f(t) \delta_t(x-v_xt, y-v_yt) \quad QED$$

**Theorem C.5** Duality for Alternative Three-dimensional Continuous Case

If a three-dimensional function, $g(x, y, t)$ has a Fourier transform which has the form of a constant-velocity translation of a two-dimensional function, $F(\omega_x, \omega_y)$, so that $G(\omega_x, \omega_y, \omega_t) = F(\omega_x v_x + \omega_y v_y + \omega_t)$, then the function $g(x, y, t)$ is given by:

$$g(x, y, t) = \frac{1}{v_x v_y} f \left( \frac{x}{v_x}, \frac{y}{v_y} \right) \delta_t \left( \frac{x}{v_x} + \frac{y}{v_y} + t \right)$$

218
Remarks: \( g(x, y, t) \) has the form of a plane of one-dimensional impulses in three-dimensional \((x, y, z)\) space; this is in contrast to previous case, where we had a line of two-dimensional impulses. The coefficient of each impulse is found from the value of \( f(x, y) \), i.e. the inverse Fourier transform of the zero-velocity function, at that point.

Proof: The two-dimensional inverse Fourier transform of \( G(\omega_x, \omega_y, \omega_t) \) is:

\[
g(x, y, t) \overset{\text{def}}{=} \iiint_{-\infty}^{+\infty} G(\omega_x, \omega_y, \omega_t)e^{+j\omega_x x}e^{+j\omega_y y}e^{+j\omega_t t} d\omega_x d\omega_y d\omega_t \tag{C.5.a}
\]

\[
= \iint_{-\infty}^{+\infty} \left[ \iint_{-\infty}^{+\infty} F(\omega_x v_x + \omega_t, \omega_y v_y + \omega_t)e^{+j\omega_x x}e^{+j\omega_y y} d\omega_x d\omega_y \right] e^{+j\omega_t t} d\omega_t \tag{C.5.b}
\]

We integrate first with respect to \( \omega_x \) and \( \omega_y \), making the substitutions: \( \omega_t' = \omega_x v_x + \omega_t \) and \( \omega_t'' = \omega_y v_y + \omega_t \). In the first integral, \( \omega_t \) is constant, so \( d\omega_t' = v_x d\omega_x \) and \( d\omega_t'' = v_y d\omega_y \):

\[
g(x, y, t) = \iint_{-\infty}^{+\infty} \left[ \iint_{-\infty}^{+\infty} F(\omega_t', \omega_t'')e^{+j(\frac{\omega_t'}{v_x} - \frac{x}{v_x})x}e^{+j(\frac{\omega_t''}{v_y} - \frac{y}{v_y})y} d\omega_t' d\omega_t'' \right] e^{+j\omega_t t} d\omega_t \tag{C.5.c}
\]

\[
= \iint_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} F(\omega_t', \omega_t'')e^{+j(\frac{\omega_t'}{v_x} - \frac{x}{v_x})x}e^{+j(\frac{\omega_t''}{v_y} - \frac{y}{v_y})y} \frac{d\omega_t'}{v_x} \frac{d\omega_t''}{v_y} \right] e^{+j\omega_t(t - \frac{x}{v_x} - \frac{x}{v_x})} d\omega_t \tag{C.5.d}
\]

The first integral is recognized as a two-dimensional inverse Fourier transform. The second is a line of one-dimensional impulses, as described in Appendix A.

\[
g(x, y, t) = \frac{1}{v_x v_y} f\left( \frac{x}{v_x}, \frac{y}{v_y} \right) \delta_t \left( \frac{x}{v_x} + \frac{y}{v_y} + t \right)
\]
Theorem C.6 Two-dimensional Discrete Case

If a two-dimensional discrete function can be expressed as a constant-velocity mod-$M$ translation of a one-dimensional function $x[m]$ which has length $M$, so that $y[m,n] = x[(m - vn) \mod M]$, for $n = 0, 1, \ldots (N - 1)$, and with $v$ a constant, then the two-dimensional discrete Fourier transform of $y[m,n]$ is given by:

$$Y[k, l] = \begin{cases} \frac{1}{N} X[k] & \text{when } \left(\frac{k}{N} + \frac{lv}{M}\right) \text{ is an integer} \\ \frac{N X[k]}{M N} \left[\frac{1 - \exp(-j 2\pi M \left(\frac{k}{N} + \frac{lv}{M}\right))}{1 - \exp(-j 2\pi \left(\frac{k}{N} + \frac{lv}{M}\right))}\right] & \text{else} \end{cases}$$

where $X[k]$ is the discrete Fourier transform of the function $x[m]$, and $k = 0, 1, \ldots (M - 1)$ and $l = 0, 1, \ldots (N - 1)$.

Remarks: Clearly, this is not a simple extension of the two-dimensional continuous result. To begin with, the discrete result explicitly requires finite-duration functions $x[m]$, and finite-duration translation. The continuous case theorem required duration for infinite duration; therefore, it will be more reasonable to compare the discrete with that obtained in the continuous domain, with temporal windowing. Indeed, the expression in square brackets has a form which is related to the discrete Fourier transform of a square pulse, i.e. the discrete analog to the sinc function. In effect, there is 'ringing' around the line $\left(\frac{k}{N} + \frac{lv}{M}\right) = 0$. This is exactly what we saw in the case of temporal windowing with a rectangular pulse in the continuous case.

Secondly, the derivation of the discrete result requires a "mod $M$" translation. By this we mean that as $x[m]$ is translated by the $vn$ term, the pixels at the edge of the image "wrap around" to the other edge, rather than just falling off the end. Perhaps the term constant-velocity rotation would be more descriptive. As shown below, there is also wrap-around in hidden in the form of $Y[k, l]$.

However, there are some special cases in which we get more familiar-looking forms. When both the velocity $v$ and the ratio $M/N$ are integers, then we get:
\[ Y[k, l] = \frac{X[k]}{N} \delta_d \left[ \frac{l}{N} + \frac{kv}{M} \right] \quad \text{C.6.2} \]

Here \( \delta_d \left[ \frac{l}{N} + \frac{kv}{M} \right] \) is a line of one-dimensional discrete impulses, as described in Appendix A, traversing the \([k, l]\) plane with constant velocity. With the definition \( r = M/N \), this result becomes:

\[ Y[k, l] = \frac{X[k]}{N} \delta_d [lr + kv] \quad \text{C.6.3} \]

This is clearly analogous to the continuous time result, especially when \( r = 1 \), i.e. when \( N = M \). The fact that the “ringing” goes away in these cases is a consequence of the periodicity involved in the derivation of the discrete Fourier transform, as will be discussed momentarily. By analogy to the continuous case, when an infinite number of constant-velocity segments have the same velocity, and line up such that they are indistinguishable from a constant-velocity function for all time, then there is no more temporal windowing effect; the sidelobes in the Fourier transforms of all the constant-velocity segments miraculously cancel each other out, and only the impulse line due to the constant-velocity function remains.

**Proof**: In order to find the discrete Fourier transform of \( y[m, n] \), first we construct a periodic sequence:

\[ \tilde{z}[m] \overset{\text{def}}{=} z[m \mod M] \]

In words, \( \tilde{z}[m] \) consists of periodic copies of \( z[m] \). Next, we construct a two-dimensional ‘traveling’, periodic sequence:

\[ \tilde{y}[m, n] = \tilde{z}[m - vn] \overset{\text{def}}{=} z[(m - vn) \mod M] \]
Now, we can find the two-dimensional discrete Fourier series for this periodic sequence:

\[
\tilde{Y}[k, l] \overset{\text{def}}{=} \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \tilde{y}[m, n] e^{-j2\pi \frac{km}{M}} e^{-j2\pi \frac{ln}{N}} \quad \text{C.6.a}
\]

\[
= \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \tilde{z}[m - vn] e^{-j2\pi \frac{km}{M}} e^{-j2\pi \frac{ln}{N}} \quad \text{C.6.b}
\]

Now, we use the substitution: \( m' = m - vn \).

\[
\tilde{Y}[k, l] = \frac{1}{MN} \sum_{n=0}^{N-1} e^{-j2\pi \frac{ln}{N}} \sum_{m'=0}^{M-1-vn} \tilde{z}[m'] e^{-j2\pi \frac{k(m'+vn)}{M}} \quad \text{C.6.c}
\]

\[
= \frac{1}{MN} \sum_{n=0}^{N-1} e^{-j2\pi n \left( \frac{k}{N} + \frac{h}{M} \right)} \sum_{m'=0}^{N-1-vn} \tilde{z}[m'] e^{-j2\pi \frac{km'}{M}} \quad \text{C.6.d}
\]

The second summation is the discrete Fourier series for \( \tilde{z}[m] \). The shifted limits of summation do not affect this result because the expression being summed is periodic in \( M \), and we are summing over one period of \( \tilde{z} \). So we have:

\[
\tilde{Y}[k, l] = \frac{1}{MN} \tilde{X}[k] \sum_{n=0}^{N-1} e^{-j2\pi n \left( \frac{k}{N} + \frac{h}{M} \right)} \quad \text{C.6.e}
\]

We evaluate this summation normally, to give:

\[
\tilde{Y}[k, l] = \frac{1}{MN} \tilde{X}[k] \left[ \frac{1 - e^{-j2\pi M \left( \frac{k}{N} + \frac{h}{M} \right)}}{1 - e^{-j2\pi \left( \frac{k}{N} + \frac{h}{M} \right)}} \right]
\]

except that when \( \left( \frac{k}{N} + \frac{h}{M} \right) \) is an integer, the denominator of this fraction is zero, making the expression undefined. In those cases, the argument of the summation becomes unity, giving:
\[
\hat{Y}[k, l] = \frac{1}{MN} \hat{X}[k] M, \quad \text{if } \left( \frac{l}{N} + \frac{kv}{M} \right) \text{ is an integer}
\]

We note from the definition of the two-dimensional discrete Fourier transform, Equation C.6.a, that \(\hat{Y}[k, l]\) has the same periodicity as \(\hat{y}[m, n]\). Specifically, \(\hat{Y}[(k + aM), (l + bN)] = \hat{Y}[k, l]\), where \(a\) and \(b\) are integers.

From Equation C.6.e, we see exactly where the periodicity arises. Due to the complex exponential term \(\exp\{k m'/M\}\), we see that \(\hat{X}[k]\) is periodic in \(k\), with period \(M\). The other complex exponential term gives rise to two periodicities. In particular, since:

\[
\exp\left\{ -j2\pi \frac{(l + aN)}{N} + \frac{(kv + bM)}{M} \right\} = \exp\left\{ -j2\pi \frac{l}{N} + \frac{kv}{M} \right\}
\]

we see that the term in square brackets in \(\hat{Y}[k, l]\) is periodic in the \(l\) dimension, with period \(N\). More interestingly though, this term is periodic in the \(k\) dimension, with period \(M/v\). As we mentioned, this term has a form which resembles the sinc function, and is aligned with the line \(\left( \frac{l}{N} + \frac{kv}{M} \right) = 0\). However, whereas the sidelobes of the sinc function die out, this function is periodic; the sidelobes diminish for the first half-period, and then increase again until after one period, there is an exact copy of the main lobe. As see have seen, the period of these “main lobes” along the \(k\) direction varies inversely with \(v\).

Now, to find the discrete Fourier transform \(Y[k, l]\), we just extract one period from the periodic \(\hat{Y}[k, l]\). That is, we multiply \(\hat{Y}[k, l]\) by a rectangular window function, which has unity value for \(k = 0, 1, \ldots (M - 1)\) and \(l = 0, 1, \ldots (N - 1)\). This gives us the general case result.

We note that the periodicity of \(\hat{Y}[k, l]\) is reflected in \(Y[k, l]\). That is, for \(v > 1\), there will be more than one period of \(\hat{Y}[k, l]\) passed by the rectangular window function. A concise way to represent this periodicity would be to replace \(\left( \frac{l}{N} + \frac{kv}{M} \right)\) by \(\left( \frac{l}{N} + \frac{(kv)\mod M}{M} \right)\). Thus, there is “wrap-around” in the discrete Fourier transform \(Y[k, l]\) just as there was in the original function, \(y[m, n]\).
Now, to prove the special case results, we make the substitution: \( \alpha = M\left(\frac{1}{N} + \frac{k\nu}{M}\right) \); then the second equation becomes:

\[
\hat{Y}[k,l] = \frac{1}{MN} \hat{X}[k] \left[ \frac{1 - e^{-j2\pi\alpha}}{1 - e^{-j2\pi\alpha/M}} \right]
\]

In the special case where \( \alpha \) is an integer, this expression is zero for all \( \alpha \neq 0 \). When \( \alpha = 0 \), the summation is the same as when \( \left(\frac{1}{N} + \frac{k\nu}{M}\right) = 0 \), which is given by the first equation. Rewriting the condition as:

\[
\alpha = \left(\frac{LM}{N} + k\nu\right)
\]

we see that \( \alpha \) is an integer, for all integers \([k, l]\), only if \( \nu \) and the ratio \( M/N \) are integers. QED.

These concepts and results are illustrated by figures in Chapter 3 of the text.
Appendix \( D \)

**Velocity-tuned Filters**

---

**Theorem D.1** Two-dimensional case

Consider a two-dimensional filter whose impulse response has the form:

\[
g(x, t) = f(t) \delta_{t}(x - vt)
\]

where \( \delta_{t}(x - vt) \) is a line of one-dimensional impulses, as described in Appendix A, traversing the \((x, t)\) plane at constant velocity. Then, convolving an arbitrary two-dimensional image, \(s(x, t)\) with \(g(x, t)\) has the effect of convolving each one-dimensional, constant-velocity "slice" of the image, \(s(x_{0} + vt, t)\), by \(f(t)\).

**Remarks**: In words, the output of this filter is composed of lines of constant velocity, \(v\), in the \((x, t)\) plane. Each such line is formed by convolving the corresponding line of the input by the one-dimensional function which was used to generate the two-dimensional filter.

The two-dimensional Fourier transform of such a filter has the form of a one-dimensional function of frequency, translated at constant velocity across the \((\omega_{x}, \omega_{t})\) plane: \(G(\omega_{x}, \omega_{t}) = F(\omega_{x}v + \omega_{t})\), where \(F(\omega_{t})\) is the Fourier transform of \(f(t)\).
Proof: The two-dimensional convolution integral is:

\[ y(x, t) = s(x, t) \ast \ast g(x, t) \]

\[ \overset{\text{def}}{=} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x - x', t - t') g(x', t') \, dx' \, dt' \]

\[ = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x - x', t - t') \delta_l(x' - v't') f(t') \, dx' \, dt' \]

Now, we integrate with respect to \( x' \) first. \( \delta_l(x' - v't') \) is a line of one-dimensional impulses, so it has the effect, upon integration, of replacing all occurrences of \( x' \) by \( v't' \). So we get:

\[ y(x, t) = \int_{-\infty}^{+\infty} s(x - v't', t - t') f(t') \, dt' \]

Now, define two functions, \( s_v(t) \) and \( y_v(t) \), which are slices of the input and output image, respectively, along lines of constant velocity, \( v \). Thus:

\[ s_v(t) \overset{\text{def}}{=} s(x, t) \bigg|_{x = x_0 + vt} = s(x_0 + vt, t) \]

Using \( D.1.c \), and the definition of \( s_v(t) \), we can evaluate \( y_v(t) \):

\[ y_v(t) \overset{\text{def}}{=} y(x_0 + vt, t) \]

\[ = \int_{-\infty}^{+\infty} s(x_0 + vt - v't', t - t') f(t') \, dt' \]

\[ = \int_{-\infty}^{+\infty} f(t') s_v(t - t') \, dt' \]
Appendix D

Velocity Tuned Filters

This, of course, is just the convolution integral, so we find:

\[ y_v(t) = s_v(t) \ast f(t) \]

**QED**

Theorem D.2 Three-dimensional Case

Consider a three-dimensional filter whose impulse response has the form:

\[ g(x, y, t) = f(t) \delta_t (x - v_x t, y - v_y t) \]

where \( \delta_t (x - v_x t, y - v_y t) \) is a line of two-dimensional impulses, as described in Appendix A, traversing \((x, y, t)\) space at constant velocity. Then convolving an arbitrary three-dimensional image, \(s(x, y, t)\), with \(g(x, y, t)\) has the effect of convolving each one-dimensional, constant-velocity "slice" of the image, \(s(x_0 + v_x t, y_0 + v_y t, t)\), by \(f(t)\).

**Remarks**: The output of this filter is composed of lines of constant velocity, \(v_x\) and \(v_y\) in \((x, y, t)\) space. Each such line is formed by convolving the corresponding line of the input by the one-dimensional function which was used to generate the three-dimensional filter.

The three-dimensional Fourier transform of such a filter has the form of a one-dimensional function of frequency, translated at constant velocity across \((\omega_x, \omega_y, \omega_t)\) space: \(G(\omega_x, \omega_y, \omega_t) = F(\omega_x v_x + \omega_y v_y + \omega_t)\), where \(F(\omega_t)\) is the Fourier transform of \(f(t)\).

**Proof**: The three-dimensional convolution integral is:

\[ y(x, y, t) = s(x, y, t) \ast \ast \ast g(x, y, t) \]

\[ = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x - x', y - y', t - t') g(x', y', t') \, dx' \, dy' \, dt' \]

\[ D.2.a \]

\[ D.2.b \]
\[ = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x - x', y - y', t - t') \delta(t - t') f(t') \, dx' \, dy' \, dt' \quad \text{D.2.c} \]

Now, we integrate with respect to \( x' \) and \( y' \). \( \delta(t - t') f(t') \) is a line of two-dimensional impulses, so it has the effect, upon integration with respect to these variables, of replacing \( x' \) by \( v_x t' \), and \( y' \) by \( v_y t' \). Thus, we get:

\[ y(x, y, t) = \int_{-\infty}^{+\infty} s(x - v_x t', y - v_y t', t - t') f(t') \, dt' \quad \text{D.2.d} \]

Now, define two functions, \( s_v(t) \) and \( y_v(t) \), which are slices of the input and output image, respectively, along lines of constant velocity, \( v \). Thus:

\[ s_v(t) \overset{\text{def}}{=} s(x, y, t) \bigg|_{\substack{x = x_0 + v_x t, \quad y = y_0 + v_y t, \quad t = t' \quad \text{D.2.e}}} \]

Using equation D.2.c, and the definition of \( s_v(t) \), we can evaluate \( y_v(t) \):

\[ y_v(t) \overset{\text{def}}{=} y(x_0 + v_x t, y_0 + v_y t, t) \quad \text{D.2.f} \]

\[ = \int_{-\infty}^{+\infty} s(x_0 + v_x t' - v_x t', y_0 + v_y t' - v_y t', t - t') f(t') \, dt' \quad \text{D.2.g} \]

\[ = \int_{-\infty}^{+\infty} f(t') s_v(t - t') \, dt' \quad \text{D.2.h} \]

This, of course, is just the convolution integral, so we find:

\[ y_v(t) = s_v(t) * f(t) \quad \text{QED} \]
Appendix D

Theorem D.3  Alternate Three-dimensional Case

Consider a three-dimensional filter whose impulse response has the form:

\[ g(x, y, t) = f(ax, by) \delta_l(ax + by + t) \]

where \( \delta_l(ax + by + t) \) is a line of one-dimensional impulses, as described in Appendix A. Then convolving an arbitrary three-dimensional image, \( s(x, y, t) \), with \( g(x, y, t) \) has the effect of convolving each two-dimensional, constant-velocity "plane" of the image, \( s(x, y, a(x - x_0) + b(y - y_0)) \), by \( f(ax, by) \).

Remarks: The output of this filter is composed of two-dimensional slices, parallel to the \((x, y)\) plane, taken at times specified by \(a\) and \(b\). Each such plane is formed by convolving the corresponding plane of the input by the two-dimensional function which was used to generate the three-dimensional filter.

When \(a = 1/v_x\), and \(b = 1/v_y\), then the three-dimensional Fourier transform of such a filter has the form of a two-dimensional function of frequency, translated at constant velocity across \((\omega_x, \omega_y, \omega_t)\) space: \(G(\omega_x, \omega_y, \omega_t) = F(\omega_x v_x + \omega_t, \omega_y v_y + \omega_t)\), where \(F(\omega_x, \omega_y)\) is the Fourier transform of \(f(x, y)\).

Proof: The three-dimensional convolution integral is:

\[ y(x, y, t) = s(x, y, t) \ast \ast \ast g(x, y, t) \]

\[ \overset{\text{def}}{=} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x' - x, y' - y, t' - t) g(x', y', t') \, dx' \, dy' \, dt' \quad  \text{D.3.a} \]

\[ = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} s(x' - x, y' - y, t' - t) \delta_l(ax' + by' + t') f(ax', by') \, dx' \, dy' \, dt' \quad  \text{D.3.c} \]
Now, we integrate with respect to $t'$. $\delta_t (ax' + by' + t')$ is a line of one-dimensional impulses, so it has the effect, upon integration with respect to these variables, of replacing $t'$ by $-(ax' + by')$. Thus, we get:

$$y(x,y,t) = \int_{-\infty}^{+\infty} s(x - x', y - y', t - ax' - by') f(ax', by') \, dx' \, dy' \quad D.3.d$$

Now, define two functions, $s_0(x,y)$ and $y_0(x,y)$, which are two-dimensional "slices" of the input and output image, respectively, taken at times specified as below:

$$s_0(x,y) \overset{\text{def}}{=} s(x,y,t) \bigg|_{t = a(x - x_0) + b(y - y_0)} = s(x,y, a(x - x_0) + b(y - y_0)) \quad D.3.e$$

Using equation $D.3.c$, and the definition of $s_0(x,y)$, we can evaluate $y_0(x,y)$:

$$y_0(x,y) \overset{\text{def}}{=} y(x,y, a(x - x_0) + b(y - y_0)) \quad D.3.f$$

$$= \int_{-\infty}^{+\infty} s(x - x', y - y', (a(x - x_0) + b(y - y_0) - ax' - by')) f(ax', by') \, dx' \, dy' \quad D.3.h$$

$$= \int_{-\infty}^{+\infty} s(x - x', y - y', (a(x - x' - x_0) + b(y - y' - y_0))) f(ax', by') \, dx' \, dy' \quad D.3.i$$

This, of course, is just the convolution integral, so we find:

$$y_0(x,y) = s_0(x,y) \ast f(x,y) \quad Q.E.D.$$

230
Appendix E

Reconstruction With Velocity-tuned Filters

Theorem E.1 Bandwidth Constraint in Two-Dimensional Case

Consider a two-dimensional constant-velocity function which is constructed from a one-dimensional function with spatial bandwidth $W$, and is then sampled at rate $f_s$. If we pass the sampled function through a velocity-tuned filter, whose filter generator function has temporal bandwidth $B_t$, then the spectral replicates of the original function will be eliminated from the Fourier transform of the sampled function if $B_t$ satisfies:

$$B_t < f_s - W |v_t - v_f|$$

Proof: With reference to Figure E.1.1, we can write the equations for the support of the first spectral replicates, and of the upper limit of the filter support:

$$f_t^r(f_x) = -v_t(f_x) + f_s$$

$$f_t^{2r}(f_x) = -v_f(f_x) + B_t$$
Figure E.1.1 Reconstruction of Constant Velocity Function with Velocity Tuned Filter
The critical points occur when \( f_z = W \) and \( f_z = -W \), where the first replicate makes its closest approach to the support of the filter. Thus, we solve the above equations for the variable \( f_z \), and imposc the conditions:

\[
\frac{f_z^*}{f_z^*} > f_z^* \bigg|_{f_z = \pm W}
\]

which gives:

\[
f_s - v_i W > -v_f (W) + B_t \quad \text{and} \quad f_s + v_i W > -v_f (-W) + B_t
\]

or equivalently:

\[
f_s + W(v_f - v_i) > B_t \quad \text{and} \quad f_s + W(v_i - v_f) > B_t
\]

which can be expressed concisely as:

\[
B_t < f_s \pm W (v_i - v_f)
\] \hspace{1cm} E.1.1

Since the minimum of two values \((a, -a)\) has a negative sign, and magnitude \(|a|\), we can write this result as:

\[
B_t < f_s - W |v_i - v_f| \quad \text{QED}
\]

**Theorem E.2** Minimum Sampling Rate in Two-dimensional Case

Consider a two-dimensional constant-velocity function which is constructed from a one-dimensional function with spatial bandwidth \( W \). If we wish to sample it and reconstruct the original function using a velocity-tuned filter of specified velocity, and relative temporal bandwidth \( B_t \), the the sampling rate must satisfy this minimum condition:
\[ f_s > 2W |v_i - v_f| \]

**Proof**: We rewrite an intermediate result of the previous theorem, to find the minimum sampling frequency needed move the spectral replicates out of the passband of a velocity-tuned filter of specified velocity. Thus, from Equation E.1.1:

\[ f_s > \pm W (v_i - v_f) + B_t \]

Now, in Section 2.3.4, we derived a condition for the minimum value of \( B_t \) needed to pass the spectrum of a constant-velocity function. This result, Equation 2.3.1, also clearly gives the condition under which a velocity-tuned reconstruction filter will pass the spectrum of the original constant-velocity function unchanged:

\[ B_t > \pm W (v_i - v_f) \]

Combining these conditions gives us four conditions on \( f_s \) which, if satisfied, guarantee both that the spectral replicates will be eliminated, and that the original spectrum will be recovered unchanged. Thus, these are the conditions which must be met if the sampled function is to be reconstructed by the velocity-tuned filter:

\[
\begin{align*}
& f_s > W(v_i - v_f) + W(v_i - v_f) \quad \text{and} \\
& f_s > W(v_i - v_f) + W(v_f - v_i) \quad \text{and} \\
& f_s > W(v_f - v_i) + W(v_i - v_f) \quad \text{and} \\
& f_s > W(v_f - v_i) + W(v_f - v_i)
\end{align*}
\]

which can be expressed concisely as:

\[ f_s > \pm 2W (v_i - v_f) \]
As in the previous theorem, we note that the last result is equivalent to the following:

\[ f_s > 2W |v_i - v_f| \]

QED

**Theorem E.3** Arbitrary Bandlimited Two-dimensional Functions

Consider a arbitrary function whose spatial bandwidth is \( W_z \), and whose temporal bandwidth is \( W_t \). If we pass the sampled function through a velocity-tuned filter, whose filter generator function has temporal bandwidth \( B_t \), then the spectral replicates of the original function will be eliminated from the Fourier transform of the sampled function if \( B_t \) satisfies:

\[ B_t < f_s - W_t - |v_f W_z| \]

**Proof**: First, from Section 2.2.4, we rearrange Equation 2.2.6 to give the minimum value of \( B_t \), which will allow the velocity-tuned filter to pass the original spectrum approximately unchanged:

\[ B_t > W_t + v_f W_z \]

With reference to Figure E.3.1, we can write the equation for the upper limit of the support of the velocity-tuned filter:

\[ f_t^u(f_z) = -v_f (f_z) + B_t \]

The critical point occurs at the points \((W_z, W_t)\) and \((-W_z, W_t)\), where the first spectral replicate makes its closest approach to the support of the filter. If the replicate is to be excluded from the passband of the filter, the following condition must be satisfied:

235
Figure E.3.1 Reconstruction of Arbitrary Bandlimited Function

With Velocity-Tuned Filter
\[ f_t^u(W_z) < f_s - W_t \quad \text{and} \quad f_t^u(-W_z) < f_s - W_t \]

which gives:

\[ (-v_f)(W_z) + B_t < f_s - W_t \quad \text{and} \quad (-v_f)(-W_z) + B_t < f_s - W_t \]

or equivalently:

\[ B_t < f_s - W_t + v_f W_z \quad \text{and} \quad B_t < f_s - W_t - v_f W_z \]

which can be expressed concisely as:

\[ B_t < f_s - W_t \pm v_f W_z \quad \text{E.3.1} \]

Again, we express this result as:

\[ B_t < f_s - W_t - |v_f W_z| \quad \text{QED} \]

**Theorem E.4** Bandwidth Constraint in Three-dimensional Case

Consider a three-dimensional constant-velocity function which is constructed from a two-dimensional function with spatial bandwidths \( W_z \) and \( W_y \), and is then sampled at rate \( f_s \). If we pass the sampled function through a velocity-tuned filter, whose filter generator function has temporal bandwidth \( B_t \), then the spectral replicates of the original function will be eliminated from the Fourier transform of the sampled function if \( B_t \) satisfies:

\[ B_t < f_s - W_z |v_z^i - v_z^f| \quad \text{and} \quad B_t < f_s - W_y |v_y^i - v_y^f| \]
Proof: First, we derive the conditions under which spectral replicates are filtered out by a velocity-tuned filter. With reference to Figure E.4.1, we see that there are four critical points; we must guarantee that none of the four corners of the first spectral replicate touch the plane which limits the support of the velocity-tuned filter.

The equations which describe the planes are:

\[ f_x v_x^i + f_y v_y^i + f_t^u = B_t \]
\[ f_x v_x^i + f_y v_y^i + f_t^l = f_s \]

We impose the condition that the support of the first replicate be above the surface of the filter support, at the four corners

\[ f_t^l (\pm W_x, \pm W_y) > f_t^u (\pm W_x, \pm W_y) \]

which gives:

\[ W_x v_x^i + W_y v_y^i + f_s > B_t + W_x v_x^l + W_y v_y^l \quad \text{and} \]
\[ W_x v_x^i - W_y v_y^i + f_s > B_t + W_x v_x^l - W_y v_y^l \quad \text{and} \]
\[ -W_x v_x^i + W_y v_y^i + f_s > B_t - W_x v_x^l + W_y v_y^l \quad \text{and} \]
\[ -W_x v_x^i - W_y v_y^i + f_s > B_t - W_x v_x^l - W_y v_y^l \]

We then add these four equations, two at a time, and solve for \( f_s \). The six resulting equations can be summarized as:

\[ B_t < f_s \pm W_x \left( v_x^i - v_x^l \right) \quad \text{and} \quad E.4.1 \]
\[ B_t < f_s \pm W_y \left( v_y^i - v_y^l \right) \quad \text{and} \]
\[ B_t < f_s \]

238
Figure E.4.1: Reconstruction of 3D Constant Velocity Function using Velocity Tuned Filter
Appendix E  Reconstruction with Velocity-Tuned Filters

Of course, the third equation is redundant for positive $W_z$ and $W_y$. Thus, we can summarize this result as:

$$B_t < f_s - W_z \left| v^i_z - v^f_z \right|$$

$$B_t < f_s - W_y \left| v^i_y - v^f_y \right|$$  \hspace{1cm} \text{QED}$$

**Theorem E.5** Minimum Sampling Rate for Three-dimensional Case

Consider a three-dimensional constant-velocity function which is constructed from a two-dimensional function with spatial bandwidths $W_z$ and $W_y$. If we wish to sample it and reconstruct the original function using a velocity-tuned filter of specified velocity, and relative temporal bandwidth $B_t$, the sampling rate must satisfy this minimum condition:

$$f_s > 2W_z \left| v^i_z - v^f_z \right|$$

$$f_s > 2W_y \left| v^i_y - v^f_y \right|$$

**Proof:** We rewrite an intermediate result of the previous theorem, to find the minimum sampling frequency needed to move the spectral replicates out of the passband of a velocity-tuned filter of specified velocity. Thus, from Equation E.4.1:

$$f_s > B_t \pm W_z \left( v^i_z - v^f_z \right)$$

$$f_s > B_t \pm W_y \left( v^i_y - v^f_y \right)$$

Now, in Section 2.3.7, we derived a condition for the minimum value of $B_t$ needed to pass the spectrum of a constant-velocity function. This result, Equation 2.3.7, also clearly gives the condition under which a velocity-tuned reconstruction filter will pass the spectrum of the original constant-velocity function unchanged:

240
Appendix E

Reconstruction with Velocity-Tuned Filters

\[ B_t > \pm W_x \left( v^i_x - v^f_x \right) \quad \text{and} \]
\[ B_t > \pm W_y \left( v^i_y - v^f_y \right) \]

We can now combine the two conditions on \( f_s \) and \( B_t \) to get four conditions on \( f_s \) which, if satisfied, guarantee that both that the spectral replicates will be eliminated, and that the original spectrum will be recovered unchanged. Thus, these are the conditions which must be met if the sampled function is to be reconstructed by the velocity-tuned filter:

\[ f_s > \pm W_x \left( v^i_x - v^f_x \right) \pm W_x \left( v^i_x - v^f_x \right) \quad \text{and} \]
\[ f_s > \pm W_x \left( v^i_x - v^f_x \right) \pm W_y \left( v^i_y - v^f_y \right) \quad \text{and} \]
\[ f_s > \pm W_y \left( v^i_y - v^f_y \right) \pm W_x \left( v^i_x - v^f_x \right) \quad \text{and} \]
\[ f_s > \pm W_y \left( v^i_y - v^f_y \right) \pm W_y \left( v^i_y - v^f_y \right) \]

These conditions can be combined to yield:

\[ f_s > \pm 2 W_x \left( v^i_x - v^f_x \right) \quad \text{and} \]
\[ f_s > \pm W_x \left( v^i_x - v^f_x \right) \pm W_y \left( v^i_y - v^f_y \right) \quad \text{and} \]
\[ f_s > \pm 2 W_y \left( v^i_y - v^f_y \right) \]

The middle condition will always be satisfied as long as the other two are, so we can summarize:

\[ f_s > 2 W_x \left| v^i_x - v^f_x \right| \quad \text{and} \]
\[ f_s > 2 W_y \left| v^i_y - v^f_y \right| \]

QED
**Theorem E.6**  Arbitrary Bandlimited Three-dimensional Functions

Consider an arbitrary three-dimensional function whose spatial bandwidths are $W_x$ and $W_y$, and whose temporal bandwidth is $W_t$. If we pass the sampled function through a velocity-tuned filter, whose filter generator function has temporal bandwidth $B_t$, then the spectral replicates of the original function will be eliminated from the Fourier transform of the sampled function if $B_t$ satisfies:

$$B_t < f_s - W_t - \left| v_x^f W_x \right| \quad \text{and}$$
$$B_t < f_s - W_t - \left| v_y^f W_y \right|$$

**Proof**: First, from Section 2.2.8, we rearrange Equation 2.2.7 to find the minimum value of $B_t$ which will allow the velocity-tuned filter to pass the original spectrum approximately unchanged:

$$B_t > W_x v_x^f + W_y v_y^f + W_t$$

With reference to Figure E.6.1, we can write the equation for the upper limit of the support of the velocity-tuned filter:

$$f_x v_x^f + f_y v_y^f + f_t^u = B_t$$

The critical point occurs at the four points $(\pm W_x, \pm W_y, f_s - W_t)$ and where the first spectral replicate makes its closest approach to the support of the filter. If the replicate is to be excluded from the passband of the filter, the following condition must be satisfied:

$$f_t^u (W_x, \pm W_y) < f_s - W_t \quad \text{and}$$
$$f_t^u (-W_x, \pm W_y) < f_s - W_t$$

242
Figure E.6.1 3D Bandwidth Requirements for Arbitrary Input

Only the first quadrant is shown
which gives:

\begin{align*}
B_t - W_z v_z^f - W_y v_y^f &< f_s - W_t \quad \text{and} \\
B_t + W_z v_z^f - W_y v_y^f &< f_s - W_t \quad \text{and} \\
B_t - W_z v_z^f + W_y v_y^f &< f_s - W_t \quad \text{and} \\
B_t + W_z v_z^f + W_y v_y^f &< f_s - W_t
\end{align*}

We then add these four equations, two at a time, and solve for \( f_s \). The six resulting equations can be summarized as:

\begin{align*}
B_t &< f_s - W_t \pm W_z v_z^f \quad \text{and} \\
B_t &< f_s - W_t \pm W_y v_y^f \quad \text{and} \\
B_t &< f_s - W_t
\end{align*}

The third condition is redundant, so we can summarize this result as:

\begin{align*}
B_t &< f_s - W_t - \left| v_z^f W_z \right| \quad \text{and} \\
B_t &< f_s - W_t - \left| v_y^f W_y \right| \quad \text{QED}
\end{align*}

Finally, as before, these two constraints on \( B \) can be used to find the minimum sampling rate needed to use a given velocity-tuned filter to reconstruction an arbitrary function from its samples:

\[ f_s > 2 W_t + 2 v_z^f W_z + 2 v_y^f W_y \]
Regularized Filters

**Theorem F.1**  Two-Dimensional Regularized Filter

Consider the functional:

\[
J = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [I - I_s]^2 \, dx \, dt + \lambda \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [\nabla_\theta I]^2 \, dx \, dt
\]

Where \( \nabla_\theta I \) is the first directional derivative of \( I = I(x, t) \), along the direction specified by \( \vec{a} = a_x \hat{x} + a_t \hat{t} \). We are given the function \( I_s = I_s(x, t) \), and want to find \( I = I(x, t) \) which minimizes \( J \). We now show that the Fourier transform of \( I \) is related to the Fourier transform of \( I_s \) by the following:

\[
H(\omega_x, \omega_t) = \frac{I(\omega_x, \omega_t)}{I_s(\omega_x, \omega_t)} = \frac{1}{1 + \frac{\lambda}{|\vec{a}|^2} \left[a_x \omega_x + a_t \omega_t\right]^2}
\]

**Remarks**: When \( \vec{a} \) lies along a constant-velocity line with velocity \( v \) in the \((x, t)\) plane, then \( \vec{a} = vt \hat{t} + t \hat{t} \), and the result becomes:

245
\[ H(\omega_x, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_x, \omega_t)}{I_s(\omega_x, \omega_t)} = \frac{1}{1 + \frac{\lambda}{(v^2 + 1)} [v \omega_x + \omega_t]^2} \]

**Proof**: The directional derivative, \( \nabla_d I \), can be found as the dot-product of the gradient of \( I \) with the unit vector in the direction of \( \vec{a} \). Thus:

\[ \nabla_d I = \nabla I \cdot \frac{\vec{a}}{|\vec{a}|} \]

\[ = \left( \frac{\partial I}{\partial x} \hat{x} + \frac{\partial I}{\partial t} \hat{t} \right) \cdot \frac{a_x \hat{x} + a_t \hat{t}}{|\vec{a}|} \]

\[ = \left( a_x \frac{\partial I}{\partial x} + a_t \frac{\partial I}{\partial t} \right) \frac{1}{|\vec{a}|} \]

Using the notation: \( I_s = \frac{\partial I(x,t)}{\partial x} \), etc, the variational principle becomes:

\[ J = \iint_{-\infty}^{+\infty} F(x, y, I, I_x, I_t) \, dx \, dt \]

\[ = \iint_{-\infty}^{+\infty} [I - I_s]^2 + \frac{\lambda}{|\vec{a}|^2} (a_t I_t + a_x I_x)^2 \, dx \, dt \]

The form of the first variation of \( J \) can be found in many references, such as Courant-Hilbert [72], pg 192. The result is:

\[ \delta J = \iint_{-\infty}^{+\infty} \delta I \left\{ F_I - \frac{\partial}{\partial x} F_{I_x} - \frac{\partial}{\partial t} F_{I_t} \right\} \, dx \, dt \]

246
To minimize $J$, we set the first variation to zero. Since $\delta J$ must be zero for all variations $\delta I$, we get the following partial differential equation, the Euler-Lagrange equation:

$$F_I - \frac{\partial}{\partial z} F_{I_z} - \frac{\partial}{\partial t} F_{I_t} = 0$$

Taking the indicated partial derivatives of $F$, and substituting into this equation yields:

$$2[I - I_s] - \frac{\partial}{\partial z} \left\{2 \frac{\lambda}{|a|^2} [a_t I_t + a_z I_z] (a_z)\right\} - \frac{\partial}{\partial t} \left\{2 \frac{\lambda}{|a|^2} [a_t I_t + a_z I_z] (a_t)\right\} = 0$$

$$[I - I_s] - \frac{\lambda}{|a|^2} [a_t I_{st} + a_z I_{sz}] (a_z) - \frac{\lambda}{|a|^2} [a_t I_{tt} + a_z I_{zt}] (a_t) = 0$$

$$I - \frac{\lambda}{|a|^2} [a_x^2 I_{xx} + 2 a_x a_t I_{xt} + a_t^2 I_{tt}] = I_s$$

This linear partial differential equation can be solved by taking the Fourier transform of both sides:

$$\hat{f}(\omega_z, \omega_t) \left\{1 + \frac{\lambda}{|a|^2} [a_x^2 \omega_z^2 + 2 a_x a_t \omega_z \omega_t + a_t^2 \omega_t^2]\right\} = \hat{I}_s(\omega_z, \omega_t)$$

Thus:

$$H(\omega_z, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_z, \omega_t)}{I_s(\omega_z, \omega_t)} = \frac{1}{1 + \frac{\lambda}{|a|^2} [a_x \omega_z + a_t \omega_t]^2} \quad \text{QED}$$

247
Theorem F.2 Two-Dimensional Case With Higher-Order Stabilizer

Consider the functional:

\[ J = \int_{-\infty}^{+\infty} \left[ I - I_s \right]^2 \, dx \, dt + \lambda \int_{-\infty}^{+\infty} \left[ \nabla_{\bar{a}} \nabla_{\bar{a}} I \right]^2 \, dx \, dt \]

Where \( \nabla_{\bar{a}} \nabla_{\bar{a}} I \) is the second directional derivative of \( I = I(x, t) \), along the direction specified by \( \bar{a} = a_x \dot{x} + a_t \dot{t} \). We are given the function \( I_s = I_s(x, t) \), and want to find \( I = I(x, t) \) which minimizes \( J \). We now show that the Fourier transform of \( I \) is related to the Fourier transform of \( I_s \) by the following:

\[ H(\omega_x, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_x, \omega_t)}{I_s(\omega_x, \omega_t)} = \frac{1}{1 + \frac{\lambda}{|\bar{a}|^4} [a_x \omega_x + a_t \omega_t]^4} \]

Remarks: When \( \bar{a} \) lies along a constant-velocity line with velocity \( v \) in the \((x, t)\) plane, then \( \bar{a} = vt \dot{i} + t \dot{t} \), and the result becomes:

\[ H(\omega_x, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_x, \omega_t)}{I_s(\omega_x, \omega_t)} = \frac{1}{1 + \frac{\lambda}{(v^2 + 1)^2} [v \omega_x + \omega_t]^2} \]

Proof: \( \nabla_{\bar{a}} \nabla_{\bar{a}} I \) can be found by applying the \( \nabla_{\bar{a}} \) operator to the form for \( \nabla_{\bar{a}} \), found in Equation G.1.2. Thus:

\[ \nabla_{\bar{a}} \nabla_{\bar{a}} I = \nabla_{\bar{a}} (\nabla_{\bar{a}} I) \]

\[ \overset{\text{F.2.1}}{=} \frac{\partial}{\partial x} (\nabla_{\bar{a}} I) \frac{a_x}{|\bar{a}|} + \frac{\partial}{\partial t} (\nabla_{\bar{a}} I) \frac{a_t}{|\bar{a}|} \]
\[
\begin{align*}
&= \frac{\partial}{\partial x} \left( a_x I_x + a_t I_t \right) \frac{1}{|a|} \frac{a_x}{|a|} + \frac{\partial}{\partial t} \left( a_x I_x + a_t I_t \right) \frac{1}{|a|} \frac{a_t}{|a|} \\
&= \left( a_x I_{xx} + a_t I_{xt} \right) \frac{a_x}{|a|^2} + \left( a_x I_{xt} + a_t I_{tt} \right) \frac{a_t}{|a|^2} \\
&= \left( a_x^2 I_{xx} + 2a_x a_t I_{xt} + a_t^2 I_{tt} \right) \frac{1}{|a|^2}
\end{align*}
\]
\[F.2.2\]

Thus, the variational principle becomes:

\[
J = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(x, y, I, I_{xx}, I_{xt}, I_{tt}) \, dx \, dt
\]

\[
= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left[ I - I_0 \right]^2 + \frac{\lambda}{|\tau|} \left[ a_x^2 I_{xx} + 2a_x a_t I_{xt} + a_t^2 I_{tt} \right]^2 \, dx \, dt
\]

The form of the first variation of \( J \) can be found in many references, such as Courant-Hilbert [72], pg 192. The result is:

\[
\delta J = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \delta I \left\{ F_I + \frac{\partial^2}{\partial x^2} F_{I_{xx}} + \frac{\partial^2}{\partial x \partial t} F_{I_{xt}} + \frac{\partial^2}{\partial t^2} F_{I_{tt}} \right\} \, dx \, dt
\]

To minimize \( J \), we set the first variation to zero. Since \( \delta J \) must be zero for all variations \( \delta I \), we get the following partial differential equation, the Euler-Lagrange equation:

\[
F_I + \frac{\partial^2}{\partial x^2} F_{I_{xx}} + \frac{\partial^2}{\partial x \partial t} F_{I_{xt}} + \frac{\partial^2}{\partial t^2} F_{I_{tt}} = 0
\]

Taking the indicated partial derivatives of \( F \), and substituting into this equation yields:

249
\[ 2 \left[ I - I_s \right] + \frac{\partial^2}{\partial x^2} \left\{ \frac{2}{|a|^4} \left[ a_z^2 I_{xx} + 2 a_x a_t I_{xt} + a_t^2 I_{tt} \right] (a_z^2) \right\} \]
\[ + \frac{\partial^2}{\partial x \partial t} \left\{ \frac{2}{|a|^4} \left[ a_z^2 I_{xx} + 2 a_x a_t I_{xt} + a_t^2 I_{tt} \right] (2 a_x a_t) \right\} \]
\[ + \frac{\partial^2}{\partial t^2} \left\{ \frac{2}{|a|^4} \left[ a_z^2 I_{xx} + 2 a_x a_t I_{xt} + a_t^2 I_{tt} \right] (a_t^2) \right\} = 0 \]

\[ [I - I_s] + \frac{\lambda}{|a|^4} \left[ a_z^2 I_{zzzz} + 2 a_x a_t I_{zxtt} + a_t^2 I_{tttt} \right] (a_z^2) \]
\[ + \frac{\lambda}{|a|^4} \left[ a_z^2 I_{zzzt} + 2 a_x a_t I_{zxttt} + a_t^2 I_{ttttt} \right] (2 a_x a_t) \]
\[ + \frac{\lambda}{|a|^4} \left[ a_z^2 I_{zztt} + 2 a_x a_t I_{ztttt} + a_t^2 I_{tttt} \right] (a_t^2) = 0 \]

\[ I + \frac{\lambda}{|a|^4} \left[ a_z^4 I_{zzzz} + 4 a_z^2 a_t I_{zxtt} + 6 a_x^2 a_t^2 I_{zxttt} + 4 a_x a_t^3 I_{ztttt} + a_t^4 I_{tttt} \right] = I_s \]

This linear partial differential equation can be solved by taking the Fourier transform of both sides:

\[ \hat{I}_s(\omega_x, \omega_t) = \hat{I}(\omega_x, \omega_t) \left\{ 1 + \frac{\lambda}{|a|^4} \left[ a_z^4 \omega_x^4 + 4 a_z^2 a_t \omega_x^2 \omega_t \right. \right. \]
\[ \left. + 6 a_x^2 a_t^2 \omega_x^2 \omega_t^2 + 4 a_x a_t^3 \omega_x \omega_t^3 + a_t^4 \omega_t^4 \right] \} \]
\[ = \hat{I}(\omega_x, \omega_t) \left\{ 1 + \frac{\lambda}{|a|^4} \left[ a_x \omega_x + a_t \omega_t \right]^4 \right\} \]

Thus:

\[ H(\omega_x, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_x, \omega_t)}{I_s(\omega_x, \omega_t)} = \frac{1}{1 + \frac{\lambda}{|a|^4} \left[ a_x \omega_x + a_t \omega_t \right]^4} \]

\[ \text{QED} \]
Theorem F.3  Three-Dimensional Case

Consider the functional:

\[ J = \int\int\int_{-\infty}^{+\infty} |I - I_s|^2 \, dx \, dy \, dt + \lambda \int\int\int_{-\infty}^{+\infty} |\nabla_\vec{a} I|^2 \, dx \, dy \, dt \]

Where \( \nabla_\vec{a} I \) is the directional derivative of \( I = I(x, y, t) \), along the direction specified by \( \vec{a} = a_x \hat{x} + a_y \hat{y} + a_t \hat{t} \). We are given the function \( I_s = I_s(x, y, t) \), and want to find \( I = I(x, y, t) \) which minimizes \( J \). We now show that the three-dimensional Fourier transform of \( I \) is related to the Fourier transform of \( I_s \) by the following:

\[ H(\omega_x, \omega_y, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_x, \omega_y, \omega_t)}{I_s(\omega_x, \omega_y, \omega_t)} = \frac{1}{1 + \frac{\lambda}{|\vec{a}|^2} [a_x \omega_x + a_y \omega_y + a_t \omega_t]^2} \]

Remarks: When \( \vec{a} \) lies along a constant-velocity line with velocity components \( v_x \) and \( v_y \) in the \((x, y, t)\) plane, then \( \vec{a} = v_x t \hat{x} + v_y t \hat{y} + t \hat{t} \), and the result becomes:

\[ H(\omega_x, \omega_t) \overset{\text{def}}{=} \frac{I(\omega_x, \omega_y, \omega_t)}{I_s(\omega_x, \omega_y, \omega_t)} = \frac{1}{1 + \frac{\lambda}{(v_x^2 + v_y^2 + 1)} [v_x \omega_x + v_y \omega_y + \omega_t]^2} \]

Proof: As in the two-dimensional case, \( \nabla_\vec{a} I \) is found as the dot-product of the gradient of \( I \) with the vector \( \vec{a} \). Thus:

\[ \nabla_\vec{a} I = \nabla I \cdot \frac{\vec{a}}{|\vec{a}|} \]

\[ = \left( \frac{\partial I}{\partial x} \hat{x} + \frac{\partial I}{\partial y} \hat{y} + \frac{\partial I}{\partial t} \hat{t} \right) \cdot \frac{a_x \hat{x} + a_y \hat{y} + a_t \hat{t}}{|\vec{a}|} \]

251
\begin{equation}
= \left( a_x \frac{\partial I}{\partial x} + a_y \frac{\partial I}{\partial y} + a_t \frac{\partial I}{\partial t} \right) \frac{1}{|a|} \tag{F.3.2}
\end{equation}

Using the notation: \( I_x = \frac{\partial I(x,t)}{\partial x} \), etc, the variational principle becomes:

\[
J = \int \int_{-\infty}^{+\infty} F(x, y, I, I_x, I_t) \, dx \, dt
\]

\[
= \int \int_{-\infty}^{+\infty} \left[ I - I_0 \right]^2 + \frac{\lambda}{|a|^2} \left[ a_t I_t + a_x I_x + a_y I_y \right]^2 \, dx \, dy \, dt
\]

The form of the first variation of \( J \) can be found in many references, such as Courant-Hilbert [72], pg 192. The result is:

\[
\delta J = \int \int_{-\infty}^{+\infty} \delta I \left\{ F_I - \frac{\partial}{\partial x} F_{I_x} - \frac{\partial}{\partial y} F_{I_y} - \frac{\partial}{\partial t} F_{I_t} \right\} \, dx \, dt
\]

To minimize \( J \), we set the first variation to zero. Since \( \delta J \) must be zero for all variations \( \delta I \), we get the following partial differential equation, the Euler-Lagrange equation:

\[
F_I - \frac{\partial}{\partial x} F_{I_x} - \frac{\partial}{\partial y} F_{I_y} - \frac{\partial}{\partial t} F_{I_t} = 0
\]

Taking the indicated partial derivatives of \( F \), and substituting into this equation yields:
\[ 2 [I - I_s] - \frac{\partial}{\partial x} \left\{ 2 \frac{\lambda}{|a|^2} [a_x I_x + a_y I_y + a_t I_t] (a_x) \right\} \\
- \frac{\partial}{\partial y} \left\{ 2 \frac{\lambda}{|a|^2} [a_x I_x + a_y I_y + a_t I_t] (a_y) \right\} \\
- \frac{\partial}{\partial t} \left\{ 2 \frac{\lambda}{|a|^2} [a_x I_x + a_y I_y + a_t I_t] (a_t) \right\} = 0 \]

\[ [I - I_s] - \frac{\lambda}{|a|^2} [a_x I_{xx} + a_y I_{xy} + a_t I_{xt}] (a_x) \\
- \frac{\lambda}{|a|^2} [a_x I_{xy} + a_y I_{yy} + a_t I_{yt}] (a_y) \\
- \frac{\lambda}{|a|^2} [a_x I_{xt} + a_y I_{yt} + a_t I_{tt}] (a_t) = 0 \]

\[ I - \frac{\lambda}{|a|^2} [a_x^2 I_{xx} + 2 a_x a_y I_{xy} + 2 a_y a_t I_{xt} + 2 a_y a_t I_{yt} + a_y^2 I_{yy} + a_t^2 I_{tt}] = I_s \]

This linear partial differential equation can be solved by taking the Fourier transform of both sides:

\[ \hat{I}_s(\omega_x, \omega_t) + \hat{I}(\omega_x, \omega_t) \left\{ 1 + \frac{\lambda}{|a|^2} \left[ a_x^2 \omega_x^2 + 2 a_x a_y \omega_x \omega_y + 2 a_x a_t \omega_x \omega_t \right. \right. \]
\[ \left. \left. + 2 a_y a_t \omega_y \omega_t + a_y^2 \omega_y^2 + a_t^2 \omega_t^2 \right] \right\} \]
\[ = \hat{I}(\omega_x, \omega_t) \left\{ 1 + \frac{\lambda}{|a|^2} \left[ a_x \omega_x + a_y \omega_y + a_t \omega_t \right]^2 \right\} \]

Thus:

\[ H(\omega_x, \omega_t) \overset{\text{def}}{=} \frac{\hat{I}(\omega_x, \omega_t)}{\hat{I}_s(\omega_x, \omega_t)} = \frac{1}{1 + \frac{\lambda}{|a|^2} \left[ a_x \omega_x + a_y \omega_y + a_t \omega_t \right]^2} \]

\[ \text{QED} \]
Theorem F.4  Fourier Transform of Regularized Filter

For filters whose Fourier transform has the form:

\[ H(\omega) = \frac{1}{1 + \lambda \omega^{2N}} \quad \text{F.4.1} \]

The inverse Fourier transform, \( h(t) \) can be described two ways. First, \( h(t) \) can be described as the convolution of a \( N \)th order Butterworth filter with its time-reversed self. (This operation is identical to the auto-correlation operation.)

\[ h(t) = b_N(t) * b_N(-t) \quad \text{F.4.2} \]

\[ b_N(t) = \mathcal{L}^{-1} \{ B(s) \} \quad \Rightarrow \quad B(s) = \frac{(\omega_c)^N}{N \prod_{k=1}^{N} (s - s_k)} \quad \text{F.4.3} \]

\[ \omega_c = \left( \frac{1}{\lambda} \right)^{\frac{1}{2N}} \]

\[ s_k = \omega_c e^{j\pi \left[ \frac{1}{2} + \frac{2k - 1}{2N} \right]} \quad k = 1, 2, \ldots 2N \quad \text{F.4.4} \]
Alternatively, \( h(t) \) can be expressed as the sum of causal and anti-casual exponentially damped sinusoids:

\[
\begin{align*}
    h(t) &= \sum_{n=1}^{N/2} 2P \exp \{s_n^r |t| \} \left[ s_n^r \cos \ s_n^i |t| - s_n^i \sin \ s_n^i |t| \right] \quad N \text{ even} \\
    h(t) &= \sum_{n=1}^{(N-1)/2} 2P \exp \{s_n^r |t| \} \left[ s_n^r \cos \ s_n^i |t| - s_n^i \sin \ s_n^i |t| \right] \\
        &\quad - P \omega_c \exp \{- \omega_c |t|\} \quad N \text{ odd}
\end{align*}
\]

where:

\[
P \overset{\text{def}}{=} \left[ (-\omega_c)^2 N \prod_{k'=1}^{N-1} \left( 1 - \cos \left( \frac{\pi}{N} k' \right) \right) \right]^{-1}
\]

\[
s_n^r = \text{Re} \ (s_n) \quad s_n^i = \text{Im} \ (s_n)
\]

**Remarks:** As pointed out by Terzopoulos [57], the \( d \)-dimensional version of this filter corresponds to the class of \( d \)-dimensional multivariate generalized spline functionals of Duchon [58], and Meinguet [59]. These functionals give rise to spline approximation solutions, and have several properties which make them attractive to the application of solving visual problems. For the \( d \)-dimensional functions \( v(\vec{x}), \vec{x} = [x_1, \ldots, x_d] \), the class of functionals is defined by:

\[
|v|_m^2 = \sum_{i_1, \ldots, i_m = 1}^d \int_{\mathbb{R}^d} \left( \frac{\partial^m v(\vec{x})}{\partial x_{i_1} \cdots \partial x_{i_m}} \right)^2 d\vec{x}
\]

\[
= \int_{\mathbb{R}^d} \sum_{j_1 + \cdots + j_d = m} \frac{m!}{j_1! \cdots j_d!} \left( \frac{\partial^m v(\vec{x})}{\partial x_{i_1}^{j_1} \cdots \partial x_{i_d}^{j_d}} \right)^2 d\vec{x}.
\]

The corresponding filter has the same form as Equation F.4.1, but the frequency variable \( \omega \) is replaced by a \( d \)-dimensional frequency vector, \( \vec{\omega} \). In this particular case, where \( d = 1 \), and \( m = N \), this functional reduces to:
\[ \int \left( \frac{\partial^N f(t)}{\partial t^N} \right)^2 dt \]

The second and third order incarnations of this stabilizing functional have been investigated for regularization of the edge-detection problem in one and two dimensions; see for example [50], [49] and [43]. Closed form expressions for the corresponding filters are derived in [50].

**Proof**: The general form of this filter, given in equation F.4.1, can be written as:

\[ H(\omega) = \frac{1}{1 + \left( \frac{\omega}{\omega_c} \right)^{2N}} \quad \text{where} \quad \omega_c = \left( \frac{1}{\lambda} \right)^{\frac{1}{2N}} \quad \text{F.4.a} \]

First, we assert that we can construct the bilateral LaPlace transform for our filter, by substituting \( s \) for \( j\omega \) in the Fourier transform expression, with the proviso that the LaPlace transform must include the \( j\omega \) axis in its region of convergence, so that the Fourier transform converges; see [73] for more details. So we have:

\[ H(s) = \frac{1}{1 + \left( \frac{s}{j\omega_c} \right)^{2N}} \quad \text{F.4.b} \]

Now, this LaPlace transform has \( 2N \) poles, evenly spaced in the complex \( s \)-plane, located at:

\[ s_k = j\omega_c (-1)^{\frac{k}{2N}} = \omega_c e^{j\pi \left[ \frac{1}{2} + \frac{2k - 1}{2N} \right]} \quad k = 1, 2, \ldots 2N \quad \text{F.4.c} \]

which gives us equation F.4.4.

At this point, we pause to establish some important symmetry properties of the pole locations given by \( s_n \).
First:

\[ s_{N+n} = \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2N + 2n - 1}{2N} \right] \right\} \]

\[ = \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2n - 1}{2N} \right] \exp \{ j\pi \} \right\} \]

\[ = (-1)s_n \quad \text{F.4.d} \]

Also:

\[ s_{N-n+1} = \omega_c \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2N - 2n + 2 - 1}{2N} \right] \right\} \]

\[ = \omega_c \exp \left\{ j\pi \left[ \frac{3}{2} + \frac{-2n + 1}{2N} \right] \right\} \]

\[ = \omega_c \exp \left\{ -j\pi \left[ \frac{1}{2} + \frac{2n - 1}{2N} \right] \right\} \]

\[ = s_n^* \quad \text{F.4.e} \]

Thus, two important symmetry properties are that for each pole located at \( s = s_n \), there is another pole at \( s = -s_n \), reflected across the origin into the the opposite half of the complex plane, and another pole at \( s = s_n^* \), reflected across the Real axis to form a complex conjugate pair.

From this equation, we can proceed in two directions, to prove the two statements of the theorem. First, we will exploit the first symmetry property to prove the first statement. We need only rewrite \( H(s) \) as given in equation F.4.b.

\[
H(s) = \frac{(j\omega_c)^{2N}}{(j\omega_c)^{2N} + s^{2N}} = \frac{(-1)^N(\omega_c)^{2N}}{\prod_{k=1}^{2N} (s - s_k)} \\
= \frac{(\omega_c)^{2N}}{(-1)^N \prod_{k=1}^{N} (s - s_k) \prod_{k=1}^{N} (s + s_k)}
\]
Now, we define \( B(s) \):

\[
B(s) = \frac{(\omega_c)^N}{N \prod_{k=1}^{N} (s - s_k)} \quad \Rightarrow \quad b(t) = \mathcal{L}^{-1}\{B(s)\}
\]

\[F.4.g\]

\( H(s) \) can now be expressed as the product of \( B(s) \) and \( B(-s) \). Thus we can take the inverse LaPlace transform of \( H(s) \), making use of the fact that multiplication in the s-domain implies convolution in the time domain, and that \( F(-s) \) and \( f(-t) \) are a LaPlace transform pair.

The inverse LaPlace transform of \( H(s) \) cannot be uniquely determined without specifying the region of convergence for the forward transform. We stated earlier that the region of convergence must include the \( j \omega \)-axis, in order that the Fourier transform exist. In addition, the region of convergence cannot contain any poles. With these two restrictions in force, the region of convergence must lie to the right of the right-most poles in the left half-plane, and to the left of the left-most poles in the right half-plane. It can be seen from equation \( F.4.c \) that the pole located at \( s = s_1 \) lies in the left half-plane, and that no other pole lies closer to the \( j \omega \)-axis that it does. From the pole symmetry exposed in equation \( F.4.d \), we see that there is also a pole located at \( s = -s_1 \); this pole then lies in the right half-plane, and no other right half-plane pole lies closer to the \( j \omega \)-axis. Thus, the region of convergence that is required in order for the Fourier transform to exist in this case is actually a strip of of the s-plane, which can be concisely described as :

\[
\text{Re}(s_1) < \text{Re}(s) < -\text{Re}(s_1)
\]

\[F.4.h\]

Since \( s_1 \) never lies on the \( j \omega \)-axis, the \( j \omega \)-axis will always be included in the region of convergence, so the assumption that we could use LaPlace transform techniques to solve this Fourier transform problem is proven to be valid.

258
Appendix F

Now, the function $B(s)$ has all of its poles in the left half-plane, and a region of convergence which is to the left of all of the poles. Therefore, its inverse LaPlace transform is guaranteed to be a stable, causal function. On the other hand, the function $B(-s)$ has all of its poles in the right half-plane, with a region of convergence to the right of the right-most pole, so its inverse LaPlace transform is guaranteed to be a stable anti-causal function, i.e. to be non-zero only for negative time. Thus $h(t)$, the inverse LaPlace transform of $H(s)$, will be non-zero for both positive and negative time.

With all of these issues cleared up, we can proceed to find the inverse LaPlace transform of $H(s)$:

$$H(s) = B(s) B(-s) \quad \Rightarrow \quad h(t) = b(t) \ast b(-t) \quad F.4.i$$

$B(s)$ is the LaPlace transform for a Butterworth filter (within a scalar). Thus, the general form of the one-dimensional regularized filter has the form of the convolution of Butterworth filters. Thus, we have proven the first form for $h(t)$, as given by equation $F.4.2$.

To prove the second statement of the theorem, we will proceed by finding a partial fraction expansion for $H(s)$, and then find the inverse LaPlace transform of each of the terms. Since all the roots are distinct, we can use a straightforward procedure:

$$H(s) = \frac{C_1}{(s - s_1)} + \frac{C_2}{(s - s_2)} + \cdots + \frac{C_{2N}}{(s - s_{2N})}$$

$$C_n = H(s) \ast (s - s_n) \bigg|_{s = s_n} \quad F.4.j$$

In other words, we cancel out the $n$th pole of $H(s)$, and evaluate the result at $s = s_n$:

$$C_n = (j \omega_c)^{2N} \left[ \prod_{k=1 \atop k \neq n}^{2N} (s_n - s_k) \right]^{-1} \quad F.4.k$$
\[ = (j \omega_c)^{2N} \left[ \prod_{k=n+1}^{2N} \left( 1 - \frac{3k}{2n} \right) \right]^{-1} \]

Now, from equation F.4c:
\[
\left( \frac{s_k}{s_n} \right) = \omega_c \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2k-1}{2N} \right] \right\} / \omega_c \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2n-1}{2N} \right] \right\}
\]
\[= \exp \left\{ j\pi \left[ \frac{2k-2n}{2N} \right] \right\} \]

Substituting:
\[C_n = (j \omega_c)^{2N} \left[ (s_n)^{2N-1} \prod_{k=n+1}^{2N} \left( 1 - \exp \left\{ j\pi \left( k - n \right) / N \right\} \right) \right]^{-1} \]  \hspace{1cm} F.4.1

Now, we simplify the exponential term, by exploiting symmetry and periodicity of the complex exponential. We split the product into two parts. The index of the first product increases from \( k = n + 1 \); the second decreases from \( k = n - 1 \). Both have \( N - 1 \) terms, so we have an extra term leftover, i.e. the \( k = (N + n) \)th term, which comes outside the product sign.

\[C_n = (j \omega_c)^{2N} \left[ (s_n)^{2N-1} \left( 1 - \exp \left\{ j\pi \left( N \right) / N \right\} \right) \right]
\times \prod_{k=n+1}^{n+N-1} \left( 1 - \exp \left\{ j\pi \left( k - n \right) / N \right\} \right) \prod_{k=n+1}^{n-N+1} \left( 1 - \exp \left\{ j\pi \left( k - n \right) / N \right\} \right) \right]^{-1} \]

Now, we make substitutions: \( k' = k - n \) in the first product, and \( k' = -(k - n) \) in the second.
\[ C_n = (j \omega_c)^{2N} \left[ (s_n)^{2N-1}(2) \prod_{k'=1}^{N-1} \left( 1 - \exp \left\{ j \frac{\pi}{N} k' \right\} \right) \prod_{k'=1}^{N-1} \left( 1 - \exp \left\{ j \frac{\pi}{N} (-k') \right\} \right) \right]^{-1} \]

We combine the products and multiply the terms in parentheses:

\[ C_n = (-1)^N (\omega_c)^{2N} \left[ 2(s_n)^{2N-1} \prod_{k'=1}^{N-1} \left( 2 - 2 \cos \left\{ \frac{\pi}{N} k' \right\} \right) \right]^{-1} \tag{F.4.m} \]

We can simplify further, using equation F.4.c:

\[ s_n^{2N-1} = (\omega_c)^{2N-1} \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2n-1}{2N} \right] 2N \right\} \exp \left\{ j\pi \left[ \frac{1}{2} + \frac{2n-1}{2N} \right] (-1) \right\} \]

\[ = (\omega_c)^{2N-1} (-1)^{N+1} s_n^a \]

Since \( s_n s_n^a = (\omega_c)^2 \):

\[ s_n^{2N-1} = (\omega_c)^{2N-1} (-1)^{N+1} \frac{(\omega_c)^2}{s_n} \]

\[ = (\omega_c)^{2N+1} (-1)^{N+1} \frac{1}{s_n} \]

Combining the last two equations gives the compact expression:

\[ C_n = P s_n \tag{F.4.n} \]

\[ P \overset{\text{def}}{=} \left[ (-\omega_c)^{2N} \prod_{k'=1}^{N-1} \left( 1 - \cos \left\{ \frac{\pi}{N} k' \right\} \right) \right]^{-1} \]

261
Appendix F

We note that $P$ is always real and negative. We also see, from equation $F.4.n$, we see that the coefficients $C_n$ will have the same symmetry properties as $s_n$.

Thus:

$$C_{N+n} = Ps_{N+n} = P - s_n = (-1)C_n \quad F.4.o$$

Similarly:

$$C_{N-n+1} = C_n^* \quad F.4.p$$

Using the result of equation $F.4.d$, i.e. that for each pole at $s = s_k$, there is another pole located at $s = -s_k$, we rewrite the partial fraction expansion given in equation $F.4.j$:

$$H(s) = \frac{C_1}{(s-s_1)} + \frac{C_2}{(s-s_2)} + \cdots + \frac{C_N}{(s-s_N)}$$

$$+ \frac{-C_1}{(s+s_1)} + \frac{-C_2}{(s+s_2)} + \cdots + \frac{-C_N}{(s+s_N)} \quad F.4.q$$

In order to find a closed-form solution for the impulse response of the filter, we need to find the inverse LaPlace transform of $H(s)$. Each of these terms in equation $F.4.q$ is the LaPlace transform of a complex exponential function. Again, we cannot uniquely determined the inverse LaPlace transform of these terms cannot be without first specifying the region of convergence the forward transforms. Since this is the same problem as discussed in the first part of the proof, we need only restate the result of that discussion, equation $F.4.h$:

$$\text{Re}(s_1) < \text{Re}(s) < -\text{Re}(s_1)$$

262
With this region of convergence, we take the inverse Laplace transform, of each term. (L and R indicate the whether the poles in question lie in the left or right half-plane.)

\[ H_n^L(s) = \frac{C_n}{s - s_n} \quad \{ \text{Re}(s) \geq \text{Re}(s_n) \} \quad \Rightarrow \quad h_n^L(t) = C_n e^{s_n t} u(t) \]

\[ H_n^R(s) = \frac{-C_n}{s + s_n} \quad \{ \text{Re}(s) < \text{Re}(-s_n) \} \quad \Rightarrow \quad h_n^R(t) = C_n e^{-s_n t} u(-t) \]

\[ u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & \text{else} \end{cases} \]

Thus:

\[ h(t) = \sum_{n=1}^{N} C_n e^{s_n t} u(t) + \sum_{n=1}^{N} C_n e^{-s_n t} u(-t) \quad \text{F.4.r} \]

This can simplified more, by noting that the first summation is non-zero while the second is zero, and vice versa.

\[ h(t) = \sum_{n=1}^{N} C_n e^{s_n |t|} \quad \text{F.4.s} \]

This is the general form of the solution. The \( C_n \) are different than they would be if we had a Butterworth filter, so this is not just a Butterworth filter with \( t \) replaced by \( |t| \).

At this point, we have enough information to find \( h(t) \), namely, we have expressions for \( s_n \) and \( C_n \). However, finding the constants is a rather tedious process; more importantly, the solution offers no insight into the behavior of \( h(t) \), other than that it is the sum of complex exponentials, with complex coefficients. It turns out that the symmetric pole locations serve to greatly simplify the task of computing the \( C_n \)'s, and lead to a more intuitive form for the solution.

First, we exploit the fact that the poles occur in complex conjugate pairs. To do this, we group the terms of \( H(s) \) into pairs, where the poles of the members of each pair are complex conjugates of each other. Thus:
\[ H_n(s) \xlongequal{\text{def}} \frac{C_n}{(s - s_n)} + \frac{C_{N-n+1}}{(s - s_{N-n+1})} = \frac{C_n}{(s - s_n)} + \frac{C_n^*}{(s - s_n^*)} \]

With the region of convergence given in equation \( F.4.h \), the inverse Laplace transform is:
\[ h_n(t) = C_n \exp\{s_n t\} u(t) + C_n^* \exp\{s_n^* t\} u(t) \]

Now, we split \( s_n \) and \( C_n \) into their real and imaginary parts:
\[ h_n(t) = C_n \exp\{(s_n^r + js_n^i) t\} u(t) + C_n^* \exp\{(s_n^r - js_n^i) t\} u(t) \]
\[ = \exp\{s_n^r t\} \left[ C_n \exp\{j s_n^i t\} u(t) + C_n^* \exp\{-j s_n^i t\} u(t) \right] \]
\[ = \exp\{s_n^r t\} 2 \text{Re}(C_n \exp\{j s_n t\}) u(t) \]
\[ = \exp\{s_n^r t\} 2 \text{Re} \left( \left[ C_n^r + j C_n^i \right] \left[ \cos s_n^i t + j \sin s_n^i t \right] \right) u(t) \]
\[ h_n(t) = 2 \exp\{s_n^r t\} \left[ C_n^r \cos s_n^i t - C_n^i \sin s_n^i t \right] u(t) \]

When \( N \) is even, each pole is paired with its complex conjugate. Then we can rewrite equation \( F.4.s \), expressing \( h(t) \) as a sum of the \( h_n(t) \) terms.
\[ h(t) = \sum_{n=1}^{N/2} C_n e^{s_n |t|} + C_n^* e^{s_n^* |t|} = \sum_{n=1}^{N/2} h_n(|t|) \]
\[ h(t) = \sum_{n=1}^{N/2} 2 \exp\{s_n^r |t|\} \left[ C_n^r \cos s_n^i |t| - C_n^i \sin s_n^i |t| \right] \]
When $N$ is odd, there is an unpaired pole corresponding to $n = \left\lfloor \frac{N+1}{2} \right\rfloor$, and thus an unpaired term in the partial fraction expansion. The pole location and coefficient are computed as before with equations F.4.c and F.4.l:

$$s_{\frac{N+1}{2}} = -\omega_c \quad C_{\frac{N+1}{2}} = -P \omega_c$$

We use the same region of convergence as before, given by equation F.4.h, to find the inverse LaPlace transform of this term:

$$H_{\frac{N+1}{2}}(s) = \frac{-P \omega_c}{s + \omega_c} \Rightarrow h_{\frac{N+1}{2}}(t) = -P \omega_c \exp\{-\omega_c t\} u(t)$$

Combining these results:

$$h(t) = \sum_{n=1}^{N/2} 2P \exp\{s_n^*|t|\} \left[ s_n^* \cos s_n^*|t| - s_n^* \sin s_n^*|t| \right] \quad N \text{ even}$$

$$h(t) = \sum_{n=1}^{(N-1)/2} 2P \exp\{s_n^*|t|\} \left[ s_n^* \cos s_n^*|t| - s_n^* \sin s_n^*|t| \right]$$

$$- P \omega_c \exp\{-\omega_c|t|\} \quad N \text{ odd}$$
Theorem G.1 General Case

If \( g(x, t) \) has can be expressed as:

\[
g(x, t) = f(x - vt - x_m(t))
\]

Then the Fourier transform of \( g(x, t) \) is given by:

\[
G(\omega_x, \omega_t) = F(\omega_x)P(\omega_x, \omega_t)
\]

where:

\[
P(\omega_x, \omega_t) = \mathcal{F}_t \{ p(t) \}
\]

\[
p(t) = \exp \{-j[\omega_x vt + \omega_x x_m(t)]\}
\]
Appendix G  

Arbitrary Motion in the Frequency Domain

**Proof**: By definition:

\[
G(\omega_z, \omega_t) \overset{\text{def}}{=} \iint_{-\infty}^{+\infty} g(x, t) e^{-j\omega_z x} e^{-j\omega_t t} \, dx \, dt
\]

\[
= \iint_{-\infty}^{+\infty} f(x - vt - x_m(t)) e^{-j\omega_z x} e^{-j\omega_t t} \, dx \, dt
\]

Making the substitution: \( x' = x - vt - x_m(t) \), we get:

\[
G(\omega_z, \omega_t) = \int_{-\infty}^{+\infty} f(x') e^{-j\omega_z x'} \, dx' \int_{-\infty}^{+\infty} e^{-j[\omega_z vt + \omega_z x_m(t)]} e^{-j\omega_t t} \, dt
\]

\[
= F(\omega_z) P(\omega_z, \omega_t)
\]

where:

\[
P(\omega_z, \omega_t) = \mathcal{F} \{ \exp \{-j[\omega_z vt + \omega_z x_m(t)]\} \} \quad QED
\]

**Theorem G.2** Sinusoidal Modulation

For the function:

\[
p(t) = \exp \{-j(\omega_z vt + A \omega_z \sin \omega_m t)\}
\]

The Fourier transform is given by:

\[
P(\omega_t) = \sum_{n=-\infty}^{+\infty} J_n(A \omega_z) \delta(\omega_t + \omega_z v + n\omega_m)
\]

267
Proof:

We define

\[ p(t) = \hat{p}(t) \exp \{-j(\omega_m t)\} \]

where: \[ \hat{p}(t) = \exp \{-j(A\omega_x \sin \omega_m t)\} \]

Now, \( \hat{p}(t) \) is periodic, with fundamental frequency equaling the modulation frequency \( \omega_m \). So we can write \( \hat{p}(t) \) as a Fourier series:

\[ \hat{p}(t) = \sum_{n=-\infty}^{+\infty} c_n \exp \{jn\omega_m t\} \]

where:

\[ c_n = \frac{\omega_m}{2\pi} \int_{\frac{n\pi}{\omega_m}}^{\frac{(n+1)\pi}{\omega_m}} \hat{p}(t) \exp \{-jn\omega_m t\} \, dt \]

\[ = \frac{\omega_m}{2\pi} \int_{\frac{n\pi}{\omega_m}}^{\frac{(n+1)\pi}{\omega_m}} \exp \{-j(A\omega_x \sin \omega_m t)\} \exp \{-jn\omega_m t\} \, dt \]

Substituting \( z = \omega_m t \), we get:

\[ c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \exp \{j(A\omega_x \sin z - nz)\} \, dz \]

Now, this last integral is the \( n \)th order Bessel function of the first kind, with argument \( -A\omega_x \), which we represent by \( J_n(-A\omega_x) \). Thus, we can write for \( \hat{p}(t) \):

\[ \hat{p}(t) = \sum_{n=-\infty}^{+\infty} J_n(-A\omega_x) \exp \{jn\omega_m t\} \]

Substituting this expression gives \( p(t) \) as:
\[ p(t) = \exp \left\{ -j (\omega_x vt) \right\} \sum_{n=-\infty}^{+\infty} J_n (-A \omega_x) \exp \{ jn\omega_m t \} \]

\[ p(t) = \sum_{n=-\infty}^{+\infty} J_n (-A \omega_x) \exp \{ j (-\omega_x v + n\omega_m) t \} \]

The Fourier transform of \( p(t) \) can easily be found from the Fourier series representation; each spectral line in the series gives rise to an impulse in the Fourier transform. Thus :

\[ P(\omega_t) = \sum_{n=-\infty}^{+\infty} J_n (-A \omega_x) \delta (\omega_t + \omega_x v - n\omega_m) \]

Finally, we replace \( n \) by \(-n\), and use a symmetry property of Bessel functions : \( J_{-n} (-\alpha) = J_n (\alpha) \), to get :

\[ P(\omega_t) = \sum_{n=-\infty}^{+\infty} J_n (A \omega_x) \delta (\omega_t + \omega_x v + n\omega_m) \]

\[ \text{QED} \]

**Theorem G.3** Sinusoidal Modulation in Two Dimensions

For the function :

\[ p(t) = \exp \left\{ -j [\omega_x a_x t + A_1 \omega_x \sin \omega_1 t] \right\} \exp \left\{ -j [\omega_y a_y t + A_2 \omega_y \sin \omega_2 t] \right\} \]

The Fourier transform is given by :

\[ P(\omega_x, \omega_y, \omega_t) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n (A_1 \omega_x) J_m (A_2 \omega_y) \delta_t (\omega_t + \omega_x a_x + n\omega_1 + \omega_y a_y + m\omega_2) \]

**Proof:**

269
Appendix G

Arbitrary Motion in the Frequency Domain

We define

\[ p(t) = \dot{p}_x(t) \dot{p}_y(t) \exp \{-j(\omega_x a_x t)\} \exp \{-j(\omega_y a_y t)\} \]

where:

\[ \dot{p}_x(t) = \exp \{-j(A_1 \omega_x \sin \omega_1 t)\} \]

\[ \dot{p}_y(t) = \exp \{-j(A_2 \omega_y \sin \omega_2 t)\} \]

Now, both \( \dot{p}_x(t) \) and \( \dot{p}_y(t) \) are periodic, with fundamental frequency equaling their respective modulation frequencies \( \omega_1 \) and \( \omega_2 \). So we can write them as Fourier series. In fact, we can use the result of Theorem G.2, to write down the series:

\[ \dot{p}_x(t) = \sum_{n=-\infty}^{+\infty} J_n(-A_1 \omega_x) \exp\{jn\omega_1 t\} \]

\[ \dot{p}_y(t) = \sum_{n=-\infty}^{+\infty} J_m(-A_2 \omega_y) \exp\{jm\omega_2 t\} \]

substituting this expression gives \( p(t) \) as:

\[ p(t) = \sum_{n=-\infty}^{+\infty} J_n(-A_1 \omega_x) \exp\{j(-\omega_x a_x + n\omega_1) t\} \]

\[ \times \sum_{m=-\infty}^{+\infty} J_m(-A_2 \omega_y) \exp\{j(-\omega_y a_y + m\omega_2) t\} \]

\[ p(t) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n(-A_1 \omega_x) J_m(-A_2 \omega_y) \]

\[ \times \exp\{j(-\omega_x a_x + n\omega_1 - \omega_y a_y + m\omega_2) t\} \]

The Fourier transform of \( p(t) \) can easily be found from the Fourier series representation; each spectral line in the series gives rise to an impulse in the Fourier transform. Thus:

\[ P(\omega_x, \omega_y, \omega_z) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n(-A_1 \omega_x) J_m(-A_2 \omega_y) \delta_1(\omega_t + \omega_x a_x - n\omega_1 + \omega_y a_y - m\omega_2) \]

270
Finally, we change the sign of \( n \) and \( m \), and use a symmetry property of Bessel functions: 
\[ J_{-n}(-\alpha) = J_n(\alpha), \]
to get:

\[
P(\omega_z, \omega_y, \omega_t) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} J_n(A_1 \omega_z) J_m(A_2 \omega_y) \delta_1(\omega_t + \omega_z a_z + n\omega_1 + \omega_y a_y + m\omega_2) \quad QED
\]

**Theorem G.4** Temporal Windowing – Two-dimensional Case

If a two-dimensional function, \( g(x, t) \) can be expressed as the product of a constant-velocity function and of a temporal window function, so that \( g(x, t) = f(x - vt)w(t) \), then the Fourier transform of \( g(x, t) \) is given by:

\[
G(\omega_z, \omega_t) = F(\omega_z)W(\omega_z v + \omega_t) \quad QED
\]

where \( W(\omega_t) \) is the one-dimensional Fourier transform of \( w(t) \).

**Remarks**: When \( w(t) = 1 \), then we have the “unwindowed” case which we originally considered. In this case, \( W(\omega_t) \) is a one-dimensional impulse function, so \( G(\omega_z, \omega_t) \) reduces to the impulse line, i.e. to the result we derived before:

\[
G(\omega_z, \omega_t) = F(\omega_z) \delta_1(\omega_z v + \omega_t)
\]

**Proof**: The two-dimensional Fourier transform of \( g(x, t) \) is:
\[ G(\omega_x, \omega_t) \overset{\text{def}}{=} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, t) e^{-j\omega_x x} e^{-j\omega_t t} \, dx \, dt \]  

\[ = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x - vt) w(t) e^{-j\omega_x x} \, dx \right] e^{-j\omega_t t} \, dt \]  

Now, we make the substitution: \( x' = x - vt \). Since \( v \) and \( t \) are constants within the first integral, \( dx' = dx \). We also define: \( \omega_t' = (\omega_x v + \omega_t) \). So we have:

\[ G(\omega_x, \omega_t) = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x') w(t) e^{-j\omega_x (x' + vt)} \, dx' \right] e^{-j\omega_t t} \, dt \]  

\[ = \int_{-\infty}^{+\infty} f(x') e^{-j\omega_x x'} \, dx' \int_{-\infty}^{+\infty} w(t) e^{-j(\omega_x v + \omega_t)t} \, dt \]  

\[ = \int_{-\infty}^{+\infty} f(x') e^{-j\omega_x x'} \, dx' \int_{-\infty}^{+\infty} w(t) e^{-j\omega_{t'} t} \, dt \]  

Now, we recognize the first integral as the Fourier transform of \( f(x) \). The second integral is the Fourier transform of \( w(t) \). So, we may write:

\[ G(\omega_x, \omega_t) = F(\omega_x)W(\omega_t') = F(\omega_x)W(\omega_x v + \omega_t) \]  

**QED**

**Theorem G.5** Temporal Windowing – Three-dimensional Case

If a three-dimensional function, \( g(x, y, t) \) can be expressed as the product of a two-dimensional constant-velocity function and of a temporal window function, so that \( g(x, y, t) = f(x - v_x t, y - v_y t) w(t) \), then the Fourier transform of \( g(x, y, t) \) is given by:
\[ G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y)W(\omega_x v_x + \omega_y v_y + \omega_t) \]

where \(W(\omega_t)\) is the one-dimensional Fourier transform of \(w(t)\).

**Remarks**: When \(w(t) = 1\), then we have the 'unwindowed' case which we originally considered. In this case, \(W(\omega_t)\) is a one-dimensional impulse function, so \(G(\omega_x, \omega_y, \omega_t)\) reduces to the impulse plane, i.e. to the result we derived before:

\[ G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y) \delta_p (\omega_x v_x + \omega_y v_y + \omega_t) \]

**Proof**: The derivation is a straightforward extension of the two-dimensional case. The three-dimensional Fourier transform of \(g(x, y, t)\) is:

\[ G(\omega_x, \omega_y, \omega_t) \equiv \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y, t) e^{-j \omega_x x} e^{-j \omega_y y} e^{-j \omega_t t} \, dx \, dy \, dt \quad G.5.a \]

\[ = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x - v_x t, y - v_y t) \omega(t) e^{-j \omega_x x} e^{-j \omega_y y} \, dx \, dy \right] e^{-j \omega_t t} \, dt \quad G.5.b \]

Making substitutions as before: \(x' = x - v_x t\), and \(y' = y - v_y t\); with \(t\) constant inside the first two integrals, \(dx' = dx\), and \(dy' = dy\). We also define: \(\omega_t' = (\omega_t + \omega_x v_x + \omega_y v_y)\). Then we get:

\[ G(\omega_x, \omega_y, \omega_t) = \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(x', y') \omega(t) e^{-j \omega_x x'} e^{-j \omega_y y'} dx' \, dy' \right] e^{-j \omega_t t} \, dt \]

\[ = \int_{-\infty}^{+\infty} f(x', y') e^{-j \omega_x x'} e^{-j \omega_y y'} dx' \, dy' \int_{-\infty}^{+\infty} \omega(t) e^{-j (\omega_t + \omega_x v_x + \omega_y v_y)t} \, dt \]

\[ = \int_{-\infty}^{+\infty} f(x', y') e^{-j \omega_x x'} e^{-j \omega_y y'} dx' \, dy' \int_{-\infty}^{+\infty} \omega(t) e^{-j \omega_t' t} \, dt \quad G.5.c \]
Again, the first integral is the two-dimensional Fourier transform of $f(x, y)$, and the second integral is the Fourier transform of $w(t)$. Thus:

$$G(\omega_x, \omega_y, \omega_t) = F(\omega_x, \omega_y)W(\omega_t') = F(\omega_x, \omega_y)W(\omega_x v_x + \omega_y v_y + \omega_t) \quad QED$$
References


