Imaging-Consistent Warping and Super-Resolution

Ming-Chao Chiang

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

1998
ABSTRACT

Imaging-Consistent Warping and Super-Resolution

Ming-Chao Chiang

Aimed at establishing a framework for applications that require sufficiently accurate warped intensity values to perform their tasks, this thesis begins by introducing a new class of warping algorithms that provides an efficient method for image warping using the class of imaging-consistent reconstruction/restoration algorithms proposed by Boult and Wolberg. We show that by coupling the degradation model of the imaging system directly into the imaging-consistent warping algorithms, we can better approximate the warping characteristics of real sensors, and this, in turn, significantly improves the accuracy of the warped intensity values.

We then present two new approaches for super-resolution imaging. This problem is chosen because it is not only useful in itself, but it also exemplifies the kinds of applications for which the imaging-consistent warping algorithms are designed. The “image-based” approach presumes that the images were taken under the same illumination conditions and uses the intensity information provided by the image sequence to construct the super-resolution image. However, when imaging from different viewpoints, over long temporal spans, or imaging scenes with moving 3D objects, the image intensities naturally vary. The “edge-based” approach, based on edge models and a local blur estimate, circumvents the difficulties caused by lighting variations. We show
that the super-resolution problem may be solved by direct methods, which are not only computationally cheaper, but they also give results comparable to or better than those using Irani’s back-projection method. Our experiments also show that image warping techniques may have a strong impact on the quality of super-resolution imaging. Moreover, we also demonstrate that super-resolution provides added benefits even if the final sampling rate is exactly the same as the original.

We also propose two new algorithms to address the issue of quantitatively measuring the advantages of super-resolution algorithms. The “OCR based” approach uses OCR as the fundamental measure. The “appearance matching and pose estimation based” approach uses appearance matching and pose estimation as the primary metric and image-quality metric as a secondary measure. We show that even when the images are qualitatively similar, quantitative differences appear in machine processing.
## Contents

List of Figures vi

List of Tables x

Acknowledgments xi

Chapter 1 Introduction 1

Chapter 2 Imaging-Consistent Restoration and Reconstruction 6

2.1 Introduction ....................................................... 7

2.2 Background and Definitions ................................. 10

2.2.1 Image Reconstruction ................................. 10

2.2.1.1 Simple Filters ........................................... 11

2.2.1.2 Cubic Convolution ............................... 12

2.2.2 Image Formation ........................................... 14

2.2.3 Deblurring and Our Sensor Model ......................... 16

2.2.4 Image Restoration ......................................... 17

2.2.5 Input Model ................................................. 18

2.3 Imaging-Consistent Reconstruction/Restoration Algorithms 19
2.3.1 A Quadratic Imaging-Consistent Algorithm Assuming a Rect PSF Filter .......................... 21
2.3.2 A Quadratic Imaging-Consistent Algorithm Assuming a Gaussian PSF Filter .......................... 24
2.3.3 An Imaging-Consistent Algorithm with 2-Piece Cubics Assuming a Rect PSF Filter .......................... 27
2.4 Proofs ................................................................. 30
  2.4.1 Derivation of $E_i$ and $E'_i$ ........................................ 30
  2.4.2 The Relationship between QRR and Cubic Convolution .... 32
2.5 Experimental Evaluation Using Real Images ......................... 33
2.6 General Error Properties of Imaging-Consistent Algorithms ........ 34
2.7 Conclusion .................................................................. 37

Chapter 3 Imaging-Consistent Warping 39
  3.1 Introduction .............................................................. 39
  3.2 Integrating Resampler ................................................... 40
  3.3 Examples and Applications ........................................... 49
    3.3.1 Stereo Matching after Lens Distortion Correction ........... 49
    3.3.2 Warping-Based Image Fusion for Polarization Computations .. 52
  3.4 Conclusion ................................................................. 57

Chapter 4 Image-Based Super-Resolution 58
  4.1 Introduction .............................................................. 59
  4.2 Image-Based Super-Resolution ......................................... 60
  4.3 Spectral Analysis .......................................................... 74
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.6</td>
<td>Conclusion</td>
<td>133</td>
</tr>
<tr>
<td>8.1</td>
<td>Conclusions</td>
<td>134</td>
</tr>
<tr>
<td>8.2</td>
<td>Future Work</td>
<td>138</td>
</tr>
<tr>
<td>Appendix A</td>
<td>Camera Calibration and Distortion Correction</td>
<td>147</td>
</tr>
<tr>
<td>A.1</td>
<td>Introduction</td>
<td>148</td>
</tr>
<tr>
<td>A.2</td>
<td>Camera Calibration</td>
<td>149</td>
</tr>
<tr>
<td>A.2.1</td>
<td>The Camera Model</td>
<td>149</td>
</tr>
<tr>
<td>A.2.1.1</td>
<td>The Four Steps Transformation from 3D World Coordinate System to 3D Camera Coordinate System</td>
<td>149</td>
</tr>
<tr>
<td>A.2.1.2</td>
<td>Equations Relating the 3D World Coordinate to the 2D Image Coordinate</td>
<td>153</td>
</tr>
<tr>
<td>A.2.2</td>
<td>Parallelism Constraints</td>
<td>154</td>
</tr>
<tr>
<td>A.2.3</td>
<td>Problem Definition</td>
<td>155</td>
</tr>
<tr>
<td>A.2.3.1</td>
<td>External Parameters</td>
<td>155</td>
</tr>
<tr>
<td>A.2.3.2</td>
<td>Internal Parameters</td>
<td>156</td>
</tr>
<tr>
<td>A.2.4</td>
<td>Calibrating Camera Using a Single View of Coplanar Points</td>
<td>156</td>
</tr>
<tr>
<td>A.2.4.1</td>
<td>Stage 1: Compute $R$, $T_x$, and $T_y$</td>
<td>156</td>
</tr>
<tr>
<td>A.2.4.2</td>
<td>Stage 2: Compute Effective Focal Length $f$, First Radial Lens Distortion Coefficient $\kappa_1$, and $T_z$</td>
<td>161</td>
</tr>
<tr>
<td>A.2.5</td>
<td>Calibrating Camera Using a Single View of Non-Coplanar Points</td>
<td>162</td>
</tr>
<tr>
<td>A.2.5.1</td>
<td>Stage 1: Compute $R$, $T_x$, $T_y$, and Uncertainty Scale Factor $s_x$</td>
<td>162</td>
</tr>
</tbody>
</table>
List of Figures

2.1 The image formation process and the relationship between restoration and reconstruction .......................................................... 14
2.2 Illustration of imaging-consistent reconstruction/restoration algorithms ................................................................. 23
2.3 Derivation of $E_i$ and $E_i'$ ........................................................................................................................................ 31
3.1 Linear approximation to the location of $m^{-1}(j)$ ....................................................................................................... 42
3.2 The integrating resampler assuming a Rect PSF filter ................................................................................................. 44
3.3 Integrating resampling example assuming QRR -.5 ................................................................................................. 46
3.4 Images for stereo example ........................................................................................................................................ 51
3.5 Images for polarization example before they are warped to the reference image ...................................................... 54
3.6 Images for polarization example after they are warped to the reference image ....................................................... 55
4.1 Super-resolution results at the input sampling rate. .................................................................................................. 63
4.2 Final results from an 8 64x64 image sequence taken by XC-77 ........................................................................... 65
4.3 Results from an 8 64x64 image sequence taken by XC-77 without de-blurring ............................................................. 66
4.4 Same results as shown in Figure 4.2 after being normalized to have the same mean value ........................................ 67
4.5 Same results as shown in Figure 4.2 after being normalized to have the same dynamic range ........................................ 68
4.6 Final results from a 32 64x64 very noisy image sequence taken by XC-999 ......................................................... 69
4.7 Final results from an 8 64x64 grayscale image sequence taken by XC-77 after being normalized to have the same dynamic range ......................................................... 71
4.8 Results from a toy-tank sequence after being normalized to have the same dynamic range ........................................ 72
4.9 Fourier spectra of our first experimental results shown in Section 4.2 ......................................................... 75
4.10 Display of \[ \log(1 + |F(u)|) \] where \(|F(u)|\) are the Fourier spectra shown in Figure 4.9 ......................................................... 76
4.11 Fourier spectra of our first experimental results shown in Section 4.2 ......................................................... 77
4.12 Results from a sequence of 5 synthetically down-sampled images ......................................................... 79
4.13 Display of \[ \log(1 + |F_{128}(u)|) \] ......................................................... 81
4.14 Display of \[ \log(1 + |F_{100}(u)|) \] ......................................................... 82
4.15 Display of \[ \log(1 + |F_{92}(u)|) \] ......................................................... 83
4.16 Display of \[ |F_{128}^S(u)| - |F_{128}^O(u)| \] ......................................................... 84
4.17 Display of \[ |F_{100}^S(u)| - |F_{100}^O(u)| \] ......................................................... 85
4.18 Display of \[ |F_{92}^S(u)| - |F_{92}^O(u)| \] ......................................................... 86

5.1 Two of the original images showing two different illumination conditions blown up by a factor of 4 ......................................................... 92
5.2 Final results from a 32 77x43 image sequence taken by XC-77 ......................................................... 93
5.3 3D edge-based example ......................................................... 95
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Edge Model</td>
<td>98</td>
</tr>
<tr>
<td>6.1</td>
<td>Super-resolution results from a sequence of 32 391x19 images taken by</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>XC-77</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Output of OCR for the text shown in Figure 6.1.</td>
<td>108</td>
</tr>
<tr>
<td>6.3</td>
<td>Super-resolution results from a sequence of 8 430x75 images taken by</td>
<td>109</td>
</tr>
<tr>
<td></td>
<td>XC-77</td>
<td></td>
</tr>
<tr>
<td>6.4</td>
<td>Output of OCR for the text shown in Figure 6.3.</td>
<td>110</td>
</tr>
<tr>
<td>6.5</td>
<td>Super-resolution results from a sequence of 8 490x140 images taken by</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>XC-77</td>
<td></td>
</tr>
<tr>
<td>6.6</td>
<td>Output of OCR for the text shown in Figure 6.5.</td>
<td>112</td>
</tr>
<tr>
<td>7.1</td>
<td>Setup used for automatic acquisition of object image sets</td>
<td>121</td>
</tr>
<tr>
<td>7.2</td>
<td>Experimental results</td>
<td>126</td>
</tr>
<tr>
<td>7.3</td>
<td>Experimental results</td>
<td>127</td>
</tr>
<tr>
<td>7.4</td>
<td>Experimental results</td>
<td>129</td>
</tr>
<tr>
<td>A.1</td>
<td>Camera geometry with perspective projection and radial lens distortion</td>
<td>150</td>
</tr>
<tr>
<td>A.2</td>
<td>The transformation from 3D world coordinate to 2D image coordinate</td>
<td>151</td>
</tr>
<tr>
<td>A.3</td>
<td>Illustration of parallelism constraints</td>
<td>154</td>
</tr>
<tr>
<td>A.4</td>
<td>Illustration of experimental setup for camera calibration using a set of</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>coplanar points</td>
<td></td>
</tr>
<tr>
<td>A.5</td>
<td>Illustration of experimental setup for camera calibration using a set of</td>
<td>163</td>
</tr>
<tr>
<td></td>
<td>non-coplanar points</td>
<td></td>
</tr>
<tr>
<td>A.6</td>
<td>Illustration of calibration pattern</td>
<td>181</td>
</tr>
<tr>
<td>A.7</td>
<td>Illustration of computation of calibration points</td>
<td>183</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Step-by-step results of running the integrating resampler on the input shown in Figure 3.3 .............................................. 48
3.2 Errors in warping-based polarization computations .................. 56
4.1 Running times of our first experiment assuming the same degradation model ................................................................. 73
4.2 Power of the Fourier transform of the images shown in Figure 4.12 . . 80
6.1 Characters divided into categories ........................................... 106
6.2 Abbreviations ..................................................................... 107
7.1 Experimental results of object image set 1 .................................. 124
7.2 Experimental results of object image set 2 .................................. 128
7.3 Experimental results of object image set 3 .................................. 128
7.4 Experimental results of object image set 4 .................................. 129
7.5 Experimental results of object image set 1 using edge maps for both training and recognition ............................................. 132
Acknowledgments

First and foremost, my deepest thanks go to my advisor, Terry Boult. His very positive influence on my personal and technical development will carry forward into my future endeavors. Without his guidance, support, understanding, and encouragement, this dissertation would not exist.

I am also greatly indebted to the other members of my committee—John Kender, Peter Allen, Shree Nayar, and Rakesh (Teddy) Kumar, Head of Media Vision Group, Sarnoff Corporation—for their valuable comments, help, encouragement, and friendship. I also wish to thank all my colleagues and friends who made my stay at Columbia University enjoyable.

Thanks also go to Department of Electrical Engineering and Computer Science, Lehigh University, for offering me the position of Visiting Scientist so that I can keep working closely with my advisor after he moved from Columbia University to Lehigh University in September, 1994, and to all my colleagues and friends who made my stay at the VAST lab pleasant.

The source for my decision to pursue doctoral studies in the area of Image Processing stems from my friendship with George Wolberg. I am greatly indebted to him for his long-standing inspiration and friendship.
I also wish to thank two best friends of mine in Taiwan, Johnson Sih and Robert Zacharias, for their help and friendship throughout the course of my master and doctorate work, and two best friends of mine here at Columbia University, Henry Lo and Conway Yee, for their friendship. I proudly share this accomplishment with them all. Thanks also go to all the people in the administrative offices, particularly Mel, who made sure all the paperwork flowed smoothly between the various offices.

Last, but not least, my very special thanks go to my parents, brother, and sisters, for their contiguous encouragement, support, patience, and love, and to my wife, Shu-Chen, for sustaining me with her understanding and everlasting love.

MING-CHAO CHIANG

COLUMBIA UNIVERSITY

May 1998
To my parents and my wife
Chapter 1

Introduction

Many computer vision tasks require that image warping be applied to the images under consideration to reshape their geometry. When the goal of warping is to produce output for human viewing, only moderately accurate intensity values are needed. In these cases, techniques using traditional bi-linear interpolation have been found sufficient. However, as a preprocessing step for vision, the precision of the warped intensity values is often important. For these problems, bi-linear interpolation may not be sufficient. The objective of this thesis is, therefore, to establish a framework that paves the way for applications that require sufficiently accurate warped intensity values to perform their tasks.

The thesis is organized as follows. Except for most of the details of the algorithms (including Figure 2.2) described in Section 2.3, Imaging-Consistent Reconstruction/Restoration Algorithms, and Section 2.4, Proofs, Chapter 2 is related work based on [Boult and Wolberg-1993]. This background is basically where my Ph.D. study began. The imaging-consistent reconstruction/restoration algorithms described in this chapter are based on [Boult and Wolberg-1993]; my implementation, however, is far more efficient. In this chapter, we begin by introducing the class of imaging-consistent
reconstruction/restoration algorithms that is fundamentally different from traditional approaches. We deviate from the standard practice that treats images as point samples. In this work, image values are treated as area samples generated by non-overlapping integrators. This is consistent with the image formation process, particularly for CCD, CID, and almost any digital camera. We show that superior results are obtained by formulating reconstruction as a two-stage process: image restoration followed by application of the point spread function (PSF) of the imaging sensor. By coupling the PSF to the reconstruction process, we satisfy a more intuitive fidelity measure of accuracy that is based on the physical limitations of the sensor. Efficient local techniques for image restoration are derived to invert the effects of the PSF and estimate the underlying image that passed through the sensor. The reconstruction algorithms derived herein are local methods that compare favorably to cubic convolution, a well-known local technique, and they even rival global algorithms such as interpolating cubic splines. Evaluations (See [Boult and Wolberg-1993]) are made by comparing their passband and stopband performances in the frequency domain as well as by direct inspection of the resulting images in the spatial domain. A secondary advantage of the algorithms derived with this approach is that they satisfy an imaging-consistency property. This means that they exactly reconstruct the image for some function in the given class of functions. Their error can be shown to be at most twice that of the “optimal” algorithm for a wide range of optimality constraints. We show in Section 2.4 that if linear interpolation is used to derive the value of the reconstruction at the pixel boundaries, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of $a = -.5$.

Chapter 3 presents a new class of warping algorithms [Chiang and Boult-1996b]
that generalizes the idea of imaging-consistent reconstruction/restoration algorithms derived in Chapter 2 to deal with image warping. Whereas imaging-consistent reconstruction/restoration algorithms assume that the degradation model is the same for both input and output; imaging-consistent warping algorithms go one step further, allowing

1. both input and output to have their own degradation model, and

2. the degradation model to vary its size for each output pixel.

The imaging-consistent warping algorithms, which we refer to as integrating resamplers, provide an efficient method for image warping using the class of imaging-consistent reconstruction/restoration algorithms. We show that by coupling the degradation model of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real sensors, which also significantly improve the accuracy of the warped intensity values.

When we are resampling the image and warping its geometry in a nonlinear manner, this new approach allows us to efficiently do both pre-filtering and post-filtering. Because we have already determined a functional form for the input, no spatially-varying filtering is needed, as would be the case if direct inverse mapping were done. The integrating resampler described herein also handles antialiasing of partial pixels in a straightforward manner.

Examples are given to show the use of the integrating resamplers described herein for two low-level computer vision applications. The first is geometric correction for images degraded by radial lens distortion, demonstrated as a preprocessing step in correlation-based stereo matching. The second is warping-based data fusion for polarization computations in small-baseline multi-camera systems. These two applications are chosen
because they exemplify the kinds of applications for which the integrating resamplers and the imaging-consistent reconstruction/restoration algorithms are designed—problems that use warped intensity values (as opposed to edge structure).

Chapters 4 and 5 present two new algorithms for enhancing image resolution from an image sequence. The “image-based” approach [Chiang and Boult-1996a, Chiang and Boult-1997c, Chiang and Boult-1997a] presumes that the images were taken under the same illumination conditions and uses the intensity information provided by the image sequence to construct the super-resolution image. When imaging from different viewpoints, over long temporal spans, or imaging scenes with moving 3D objects, the image intensities naturally vary. The “edge-based” approach [Chiang and Boult-1997b, Chiang and Boult-1997a], based on edge models and a local blur estimate, circumvents the difficulties caused by lighting variations. The approaches we propose herein both use the imaging-consistent warping algorithms as the underlying resampling algorithm. We show that the super-resolution problem may be solved by a direct method, which is fundamentally different from the iterative back-projection approaches proposed in the previous work. Results of our experiments show that the direct method we propose herein is not only computationally cheaper, but it also gives results comparable to or better than those using back-projection. Our experiments also show that image warping techniques may have a strong impact on the quality of image resolution enhancement, which have been completely ignored by earlier research on super-resolution. Moreover, we also demonstrate that super-resolution provides added benefits even if the final sampling rate is exactly the same as the original.

Chapters 6 and 7 are devoted to quantitative measurement of super-resolution imaging. The OCR-based approach uses OCR as the fundamental measure. The appear-
ance matching and pose estimation based approach uses appearance matching and pose estimation as the primary metric and image-quality metric as a secondary measure. We show that even when the images are qualitatively similar, quantitative differences appear in machine processing.

Chapter 8 presents the conclusions and the future work, with the emphasis on issues that need to be addressed before the super-resolution algorithms described herein are robust enough for general use in applications.

The purpose of Appendix A is threefold. First, we review a two-stage algorithm for camera calibration using off-the-shelf TV cameras and lenses [Tsai-1986, Tsai-1987, Chiang and Boult-1995]. This review is included primarily for its use in Chapter 3 and Chapter 6 to calibrate two cameras which have noticeable radial lens distortion. Second, we introduce an algorithm for automatically computing the image coordinates of the calibration points from the image of a calibration pattern. Third, we present an imaging-consistent algorithm for “unwarping” radial lens distortions once the underlying camera model is determined. Examples are given to show the use of these algorithms. Details of the implementation of the two-stage calibration algorithm can be found in [Chiang and Boult-1995].
Chapter 2

Imaging-Consistent Restoration and Reconstruction

Except for most of the details of the algorithms (including Figure 2.2) described in Section 2.3, Imaging-Consistent Reconstruction/Restoration Algorithms, and Section 2.4, Proofs, this chapter is related work based on [Boult and Wolberg-1993]. This background is basically where my Ph.D. study began. The imaging-consistent reconstruction/restoration algorithms described in this chapter are based on [Boult and Wolberg-1993]; my implementation, however, is far more efficient. In this chapter, we begin by introducing the class of imaging-consistent reconstruction/restoration algorithms that is fundamentally different from traditional approaches. We deviate from the standard practice that treats images as point samples. In this work, image values are treated as area samples generated by non-overlapping integrators. This is consistent with the image formation process, particularly for CCD, CID, and almost any digital camera. We show that superior results are obtained by formulating reconstruction as a two-stage process: image restoration followed by application of the point spread function (PSF) of the imaging
sensor. By coupling the PSF to the reconstruction process, we satisfy a more intuitive fidelity measure of accuracy that is based on the physical limitations of the sensor. Efficient local techniques for image restoration are derived to invert the effects of the PSF and estimate the underlying image that passed through the sensor. The reconstruction algorithms derived herein are local methods that compare favorably to cubic convolution, a well-known local technique, and they even rival global algorithms such as interpolating cubic splines. Evaluations (See [Boult and Wolberg-1993]) are made by comparing their passband and stopband performances in the frequency domain as well as by direct inspection of the resulting images in the spatial domain. A secondary advantage of the algorithms derived with this approach is that they satisfy an imaging-consistency property. This means that they exactly reconstruct the image for some function in the given class of functions. Their error can be shown to be at most twice that of the “optimal” algorithm for a wide range of optimality constraints. We then show in Section 2.4 that if linear interpolation is used to derive the value of the reconstruction at the pixel boundaries, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of \( a = -0.5 \).

### 2.1 Introduction

Digital image reconstruction refers to the process of recovering a continuous image from its samples. This problem is of fundamental importance in digital image processing, particularly in applications requiring image resampling such as image warping, correction for geometric distortions, and image registration. Its role in these applications is to furnish a spatial continuum of image values from discrete pixels so that the input
image may be resampled at any arbitrary position, even those at which no data was originally supplied. Despite the great flurry of activity in reconstruction, the subject remains open to new solutions designed to address the tradeoff between reconstruction accuracy and computational complexity. The objective of this chapter is to present a new class of algorithms for image reconstruction/restoration [Boult and Wolberg-1993, Chiang and Boult-1996b] that is fundamentally different from traditional approaches.

Whereas reconstruction simply derives a continuous image from its samples, restoration attempts to go one step further. It assumes that the underlying image has undergone some degradation before sampling, and so it attempts to estimate that continuous image from its corrupted samples. Restoration techniques must therefore model the degradation and invert its effects on the observed image samples. In our application, we consider a limited class of degradation models motivated by point spread functions of commonly available imaging devices.

The use of image restoration techniques permits the work presented in this chapter to achieve image reconstruction in a fundamentally different way than traditional approaches. This approach, which we refer to as imaging-consistent reconstruction, formulates reconstruction as a two-stage process: functional image restoration followed by blurring according to the sensor model. This approach is in the spirit of the work of [Huck et al.-1991] where it is argued that sampling and image formation should be considered together. Imaging-consistent algorithms directly combine knowledge of image formation and sampling into the reconstruction/restoration process. The way that knowledge is used, however, is quite different from [Huck et al.-1991].

Imaging-consistent algorithms treat the image values as area samples; that is, each sample is a weighted integral. The weighting function is the point spread function of the
imaging sensor. Multi-pixel blur is a lens artifact, not a sensor artifact, and it will be handled in a post-processing stage. Thus, we assume a “single-pixel” PSF and use this to derive a functional approximation to the deblurred (i.e., restored) image. This permits the restoration to have local support. We then blur the functional restoration by the PSF to obtain a reconstruction. By coupling the PSF to the reconstruction process, we satisfy a more intuitive fidelity measure of accuracy that is based on the physical limitations of the sensor. The result is more accurate in the sense that it is the exact reconstruction for some input function which, given the sensor model, would also produce the measured image data.

The imaging-consistent reconstruction algorithms derived herein are local methods that compare favorably to cubic convolution, a well-known local technique, and they even rival global algorithms such as interpolating cubic splines. We show in Section 2.4 that if linear interpolation is used to derive the value of the reconstruction at the pixel boundaries, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of $a = -0.5$. Evaluations are made by comparing their passband and stopband performances in the frequency domain, using an error criteria defined in [Park and Schowengerdt-1982], as well as by direct inspection of the resulting images in the spatial domain. A secondary advantage of the algorithms derived with this approach is that they satisfy an imaging-consistency property, which means that they exactly reconstruct an image for some function in an allowable class of functions. Their error is at most twice that of the “optimal” algorithm for a wide range of optimality constraints.

This chapter is organized as follows. Section 2.2 reviews previous work in image reconstruction, introduces the image formation process, and motivates the use of image
restoration for reconstruction. That section describes why they should be unified into a common formulation as well as the class of allowable functions that the restoration stage can estimate. The actual reconstruction/restoration algorithms are presented in Section 2.3. Finally, conclusions and future work are discussed in Section 2.7.

2.2 Background and Definitions

In this section, we begin with a brief review of previous work in image reconstruction based on [Wolberg-1990]. More details can be found in [Wolberg-1990]. Then, we introduce the image formation process and motivate the use of image restoration for reconstruction. We describe why they should be unified into a common formulation. Finally, the class of functions that the restoration stage can estimate are specified.

2.2.1 Image Reconstruction

Image reconstruction has received much attention in the literature [Andrews and Hunt-1977, Jain-1989, Pratt-1990]. It is well known that the sinc function is the “ideal” reconstruction (in the spatial domain) assuming a band-limited signal and sufficient sampling. There are, however, several properties of this filter that are not desirable. Since edges constitute high frequencies, and since the basis functions are themselves oscillatory (i.e., sine waves), fast edge transitions will be distorted into smooth intensity changes with persisting ringing in the vicinity. Generally, it is desired to reduce this ringing. A second difficulty with sinc interpolation is that the sinc function has infinite extent, making it impractical for digital images of finite dimensions. As a result, approximations to the “ideal” reconstructed image are sought by all practical implementations. Popular recon-
struction algorithms such as linear interpolation, cubic convolution, and cubic spline only approximate the “ideal” sinc reconstruction [Park and Schowengerdt-1983].

The problem of approximate reconstruction of a non-bandlimited image is fundamentally different from the exact reconstruction of a bandlimited image. In these cases, the traditional analysis of aliasing and truncation errors is seldom used because the expected error is often unbounded. Fortunately, the actual reconstructed images do not tend to reflect such pessimistic estimates. The problem with aliasing and truncation errors is that they are not closely related to the visual fidelity of the reconstructed image. Instead, some fidelity measure must be introduced to predict and control image distortion. See [Oakley and Cunningham-1990, Park and Schowengerdt-1983] for details. To date, the study of adequate fidelity measures for image reconstruction that incorporate the degradation due to imaging sensors is lacking.

2.2.1.1 Simple Filters

The image processing and computer graphics literature is replete with algorithms that attempt to perform reconstruction for samples taken from bandlimited images. The simplest method is nearest neighbor interpolation, where each output pixel is assigned the value of the nearest sample point in the input image. This method, also known as a sample-and-hold function, is often found in frame-buffer hardware for real-time magnification using pixel replication. For large scale factors, this leaves the output very jagged.

A more popular solution is to use bi-linear interpolation. This is implemented separably by fitting a line between each successive pixel in each row. That result is then fitted with lines between successive pixels in each column. This effectively fits a bi-linear patch to each input pixel. Bi-Linear interpolation yields reasonable results over a
large range of scale factors. Excessive magnification, however, often leaves evidence of blockiness at the edges of the magnified input pixels. This corresponds to sharp changes in the slope of neighboring patches.

Smother results are possible with higher order interpolation functions. By appealing to desirable filter responses in the frequency domain, many sources have derived cubic reconstruction algorithms. Functions of higher order often do not warrant their additional cost. In addition, they can potentially introduce excessive spatial undulations into the image. Therefore, reconstruction filters are typically not modeled by polynomials of degree greater than three. Interpolating cubic splines have been found to yield superior results. The only drawback to this method, though, is that it is a global method. Much effort has been devoted to deriving local solutions which require far less computational effort, while retaining the same level of accuracy. Cubic convolution and 2-parameter cubic filters were derived with this in mind.

2.2.1.2 Cubic Convolution

Cubic convolution is a third-degree interpolation algorithm originally suggested by Rifman and McKinnon [Rifman and McKinnon-1974] as an efficient approximation to the theoretically optimum sinc interpolation function. Its interpolation kernel is derived from constraints imposed on the general cubic spline interpolation formula to share a truncated appearance of the sinc function. The kernel is composed of piecewise cubic polynomials defined on the unit subintervals $(-2, -1)$, $(-1, 0)$, $(0, 1)$, and $(1, 2)$. Outside the interval $(-2, 2)$, the interpolation kernel is zero. As a result, each interpolated point is a weighted sum of four consecutive input points. This has the desirable symmetry property of retaining two input points on each side of the interpolating region. It gives rise to a symmetric,
space-invariant, interpolation kernel of the form

\[
h(x) = \begin{cases} 
(a + 2)|x|^3 - (a + 3)|x|^2 + 1 & 0 \leq |x| < 1 \\
|a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| < 2 \\
0 & 2 \leq |x| 
\end{cases}
\]  

(2.1)

This expression yields a family of solutions, with \(a\) allowed to be a free parameter controlled by the user. Additional knowledge about the shape of the desired result may be imposed to yield bounds on the value of \(a\). The heuristics applied to derive the kernel are motivated from properties of the ideal reconstruction filter, the sinc function. By requiring \(h\) to be concave upward at \(|x| = 1\), and concave downward at \(x = 0\), we have

\[
h(0) = -2(a + 3) < 0 \quad \Rightarrow \quad a > -3
\]

\[
h(1) = -4a > 0 \quad \Rightarrow \quad a < 0
\]

Bounding \(a\) to values between \(-3\) and \(0\) makes \(h\) resemble the sinc function. In [Rifman and McKinnon-1974], the authors use the constraint that \(a = -1\) in order to match the slope of the sinc function at \(x = 1\). This choice results in some amplification of the frequencies at the high-end of the passband. As stated earlier, such behavior is characteristic of image sharpening.

Other choices for \(a\) include \(-.5\) and 
-.75. Keys selected \(a = -.5\) by making the Taylor series approximation of the interpolated function agree in as many terms as possible with the original signal [Keys-1981]. He found that the resulting interpolating polynomial will exactly reconstruct a second-degree polynomial. Finally, \(a = -.75\) is used to set the second derivatives of the two cubic polynomials in \(h\) to 1 [Simon-1975]. This allows the second derivative to be continuous at \(x = 1\).

Of the three choices for \(a\), the value \(-1\) is preferable if visually enhanced results are desired. That is, the image is sharpened, making visual detail perceived more readily.
However, the results are not mathematically precise, where precision is measured by the order of the Taylor series. To maximize this order, the value $a = -0.5$ is preferable.

In [Maeland-1988], Maeland showed that at the Nyquist frequency, the spectrum attains a value which is independent of the free parameter $a$. The value is equal to $(48/\pi^4)f_s$, while the value at the zero frequency is $f_s$. This result implies that adjusting $a$ can alter the cutoff rate between the passband and stopband, but not the attenuation at the Nyquist frequency. In comparing the effect of varying $a$, Maeland points out that cubic convolution with $a = 0$ is superior to the simple linear interpolation method when a strictly positive kernel is necessary. The role of $a$ has also been studied in [Park and Schowengerdt-1983], where a discussion is given on its optimal selection based on the frequency content of the image.

### 2.2.2 Image Formation

Image formation is generally described as a sequence of filtering operations. We briefly review our model here, as depicted in Figure 2.1 (disregarding the feedback loops for the moment). More details can be found in [Andrews and Hunt-1977, Jain-1989, Pratt-1990]. Also, see [Huck et al.-1991], where sampling and image formation are treated together.

![Image Formation Diagram](image.png)

Figure 2.1: The image formation process and the relationship between restoration and reconstruction.
Let $f(x, y)$ be the intensity distribution of a scene at the front aperture of a lens. That distribution is acted upon by $h_1(x, y)$, the blurring component of the lens, yielding $f_1(x, y)$. A geometric distortion function $h_2(x, y)$ may be added to the image by the lens to yield image $f_2(u, v)$. For instance, lens aberrations may cause pincushion or barrel effects in the image. Although the blurring and geometric distortions induced by a real lens are not necessarily decoupled in this manner, this approximate model lends itself to conceptual and computational simplifications. If we presume that the blur is small and not dominated by depth-of-field effects, this allows us to replace a spatially-varying point spread function with a cascade of two simpler components: a spatially-invariant blur and a geometric warp. We choose to model it this way because it lends itself to conceptual simplification. A similar model was used by others for motion-blur and lens aberrations [Robbins and Huang-1972, Sawchuk-1974]. This serves to interchange spatially varying point spread functions (SVPSF) with a cascade of two simpler components: a spatially invariant blur and a warp. Recent work in efficient geometric transformations [Catmull and Smith-1980, Smith-1987, Wolberg and Boult-1989], as well as spatially invariant blur, permits cost-effective solutions to the general linear spatially varying (LSV) problem posed here.

At this point, $f_2$ strikes the image sensor where it undergoes additional blurring by a point spread function $h_3(u, v)$ to generate image $f_3(u, v)$. This blurring reflects the limitations of the sensor to accurately resolve each point without the influence of neighboring points. We choose to use a simple model wherein this blurring takes place within one pixel because for CCD and CID cameras, the physical boundaries between photo-sites generally allow only insignificant charge transfer between pixels. Thus, we ignore charge bleeding and crosstalk. This model of degradation proves to be important
for deriving local restoration algorithms later in this chapter.

Image $f_3$ undergoes spatial sampling as it hits the discrete photosites in a CCD or CID camera. In tube or vidicon cameras, there is spatial sampling (by the focus spot) in the vertical direction. The combination of $h_3$ with sampling is known as area sampling. It reflects the finite size of the sampling area. If $h_3$ is taken to be an impulse, then we have point sampling.

This is an ideal concept that is often assumed to be true for theoretical considerations, but is generally not true in practice. In either case, intensities in the sampled image $I_s$ are now defined only for integer values of $u$ and $v$. The digital image $I(u,v)$ is obtained via an analog-to-digital converter that quantizes the samples of $I_s$. This completes the image formation process, leaving $I(u,v)$ as the starting point for subsequent processing, including image reconstruction and restoration.

Traditionally, the area sampling PSF is folded into the lens PSF. However, in this model, the geometric distortion $h_2$ makes this a spatially-varying process. Thus, in this chapter, we only concern ourselves with inverting the blurring function $h_3$. The lens blur $h_1$ may be dealt with using traditional techniques. Partial compensation for $h_2$ was considered in [Boult and Wolberg-1992].

### 2.2.3 Deblurring and Our Sensor Model

There is a considerable body of work dealing with linear space-invariant (LSI) systems [Pavlidis-1982, Pratt-1990]. We assume that $h_3$ is spatially invariant. For convenience, we drop the subscripts for $f$ and $h$. The LSI systems we consider in this chapter are almost ideal imagers. In particular, we consider systems where the PSF has spatial extent equal to or smaller than the inter-sample distance. This accounts for the sensor
blurring due to non-overlapping area samplers with non-negligible sampling sizes, e.g., the electron beam in vidicons, CCD photosites, and sampling apertures in microdensitometers. Although there are actually some inter-pixel effects in CCD cameras and other discrete photosite devices, these are taken to be secondary effects compared to area sampling. Since all physically realizable imaging systems must have PSFs of finite spatial extent, point sampling—the basis for most reconstruction algorithms—is not physically realizable. Researchers have developed complex models of CCD blurring in terms of MTFs [Peter D. Burns-1990, Schreiber-1986]. Unfortunately, dealing with multi-pixel blur increases the computational complexity because the algorithm is no longer local.

In this chapter, we assume that a model of the PSF exists and consider only two rather simple models: a Rect filter and a Gaussian filter approximated by a cubic B-spline [Wolberg-1990]. According to the model developed by Burns, the Gaussian approximation is more realistic. However, as you can see in [Boult and Wolberg-1993, Figure 5], the quality of reconstruction is quite good for both.

2.2.4 Image Restoration

Given that we know $I$ and $h$ ($h_2$), we seek to solve for $f$ ($f_2$). This problem, known as image restoration, is of considerable interest in image processing. Restoration algorithms are used to invert the degradation that enters into the image formation process. It leaves us with a functional form for $f$ that enables us to perform image resampling at any desired location. We thus seek to derive an efficient restoration algorithm that inverts the point spread function of imaging sensors. The method can be guaranteed to be local as long as the PSFs have a one-pixel extent. This is a reasonable assumption for many common imaging systems.
There is an interesting relationship between reconstruction and restoration, as illustrated in Figure 3. While both processes start from the image samples in \( I \), reconstruction limits itself to the problem of deriving the continuous function \( f_3 \). Restoration attempts to estimate the original input function \( f_2 \). Obviously, the two problems of reconstruction and restoration are related, and the latter is more difficult. With reasonable assumptions, exact reconstruction of \( f_3 \) is at least theoretically possible. On the other hand, exact reconstruction of \( f_2 \) requires considerably more tenuous assumptions. Even given the necessary assumptions, the problem of restoration is still considerably more difficult.

### 2.2.5 Input Model

Before we can discuss the restoration algorithms, we must specify the class of allowable functions. The most elementary constraints on \( f \) are integrability of the product of \( f \) and the PSF and non-negativity. A reconstruction algorithm based on this model does not need to make the type of assumptions traditionally imposed. That is, it need not assume \( f \) is bandlimited, continuous, or even finite. However, to get practical models, we may wish to heuristically impose more constraints on the model. Thus, we define the class, \( F_0 \), as the space of continuous functions with a local analytic expansion almost everywhere and a bounded second derivative almost everywhere. We also define a second class, \( F_1 \), as the space of continuously differentiable functions with a local analytic expansion almost everywhere and a bounded second derivative almost everywhere. The almost everywhere statements allow, on sets of measure zero, discontinuities in the second derivative. We further restrict our implementation by assuming that the number of such discontinuities is less than \( N + 1 \) where \( N \) is the number of samples.
Due to the bounded second derivative constraint, there is an implicit limit to the amplitude of high frequency components, and an implicit bandlimit to the allowable functions in the class. We find that this type of limit seems more intuitive, as it does not require us to specify a cut-off frequency. Instead of defining a passband and a stopband, it says that the function cannot have “too sharp a turn”, a concept that naturally combines frequency and amplitude.

2.3 Imaging-Consistent Reconstruction/Restoration Algorithms

In this section, we develop the mathematics for new image reconstruction/restoration methods, which we refer to as imaging-consistent reconstruction/restoration algorithms. We say that an algorithm has the imaging-consistent property if it yields an exact reconstruction for some allowable input. These are, in turn, defined by cascading the PSF model of the imaging system with the assumed mathematical model of input images. Thus, the reconstruction is exactly consistent with some input which is indistinguishable from the actual input given the (degraded) image. As we shall see later, algorithms that satisfy this property enjoy good error characteristics for many definitions of error.

We present only one dimensional image models because higher dimensions are treated separably. Non-separable multidimensional filters can also be defined. To simplify our discussions, we use the following conventions. Let us denote the pixel values by $V_i$ and the pixel boundaries as $k_i$ with regular spacing $m$. For our algorithms, the

\footnote{Note that due to the change of the coordinate system, the algorithms described in this section are different from those given in [Boult and Wolberg-1993]. This change was made to facilitate the understanding of the approaches. It also made the proof given in Section 2.4 easier.}
intensity value $V_i$ is centered at $p_i = (k_i + k_{i+1})/2$. It is convenient to define $p_i = im$. Accordingly, $k_i = (i - 1/2)m$. It is also convenient to define $x = (t - p_k)/m$ as $t$ varies over a pixel. However, by a pixel, we mean either a pixel over the interval $[k_i, k_{i+1}]$ or a pixel over the interval $[p_i, p_{i+1}]$. This distinction is important for the restoration and reconstruction algorithms derived herein. For the restoration algorithms, $-1/2 \leq x \leq 1/2$; for the reconstruction algorithms, $0 \leq x \leq 1$. Note that two coordinate systems are used for the algorithms derived herein. One is the global coordinate system (denoted $t$) that spans an entire scanline; the other is the local coordinate system (denoted $x$) that varies over a single pixel. In most cases, we do not explicitly deal with the end conditions. However, since the algorithms are all local, the reader’s choice for the end conditions has only local effect.

Furthermore, throughout this chapter, we are assuming that the degradation models are identical for both input and output. This restriction will be removed in the next chapter where we generalize the idea of the imaging-consistent reconstruction/restoration algorithms, allowing the degradation models to vary its size for each output pixel. Also, we even allow the degradation model to vary for input and output.

We now describe the following algorithms:

1. A quadratic imaging-consistent algorithm assuming a Rect filter for the PSF. Note that the Rect function is defined to be a constant value over the specified spatial interval.

2. A quadratic imaging-consistent algorithm assuming a cubic B-spline approximation to a Gaussian PSF.

3. An imaging-consistent algorithm with two cubic pieces per pixel assuming a Rect
filter for the PSF.

We have tried numerous other methods and found these to be among the simplest and the best. Readers are encouraged to try and tailor their favorite local PSF and generate their own imaging-consistent reconstruction algorithms.

2.3.1 A Quadratic Imaging-Consistent Algorithm Assuming a Rect PSF Filter

Once our definition of information (area samples) is accepted, probably the simplest method to consider is an integrable interpolatory quadratic method. For a Rect PSF filter, this is very easy to derive. Again, we are assuming centered pixels. To ensure that the function is in $F_0$, we need continuity, but we also desire a local method. Hence, we define the value of the reconstruction at the pixel boundaries $k_i$ and $k_{i+1}$, which we refer to as $E_i$ and $E_{i+1}$. Any method of reconstruction could be used to compute these values though our examples only include cubic convolution. Note that the resulting function is in $F_0$ and “interpolates” the data in the information-based complexity sense [Traub et al.-1988].

Given the value at the edges of the pixel, an additional constraint that the integral across the pixel must equal $V_i$ gives exactly three constraints. They are

$$q_k(-1/2) = E_i, \quad q_k(1/2) = E_{i+1}, \quad \int_{-1/2}^{1/2} q_k(x) \, dx = V_i.$$  

From this, one can determine the quadratic polynomial to be

$$q_i(x) = \frac{1}{4} \left(12(E_i + E_{i+1} - 2V_i)x^2 - 4(E_i - E_{i+1})x - (E_i + E_{i+1} - 6V_i)\right)$$  

(2.2)

where $-1/2 \leq x \leq 1/2$. The integral of this quadratic over the interval $k_i$ to $k_{i+1}$ is exactly $V_i$. Using cubic convolution to derive $E_i$ and $E_{i+1}$ (See Section 2.4.1 for
As shown in Section 2.2.2, the parameter $a$ is generally in the range $[-3, 0]$ in order to make the kernel resemble the sinc function [Park and Schowengerdt-1983]. This then defines the quadratic restoration algorithm using a Rect PSF and a free parameter $a$, where $a$ is the value of the parameter used for determining $E_i$ and $E_{i+1}$. We abbreviate this as QRsR $a$.

The above restoration method is always defined, though at times it produce restorations with negative values. We note that the value of the second derivative on a segment is simply

$$
\frac{d^2 q_i(x)}{dx^2} = 6(E_i + E_{i+1} - 2V_i)
= \frac{1}{4} (3aV_{i-2} + 12V_{i-1} - (6a + 24)V_{i} + 12V_{i+1} + 3aV_{i+2}).
$$

In general, the maximum value of the second derivative is on the order of the maximum step change in image intensity values.

To define the reconstruction algorithm, we simply blur the resulting restoration by a Rect filter of the same size. One can derive a functional form for this that results in one cubic polynomial that spans from the center of one input pixel to the next. Symbolically,

$$Q_i(x) = \int_{x-1/2}^{1/2} q_i(z) \, dz + \int_{-1/2}^{x-1/2} q_{i+1}(z) \, dz
= (E_{i+2} - E_i - 2(V_{i+1} - V_i)) x^3
+ (2E_i - E_{i+1} - E_{i+2} + 3(V_{i+1} - V_i)) x^2
+ (E_{i+1} - E_i) x + V_i \quad (2.3)$$
where $0 \leq x \leq 1$. We refer to the resulting (globally) continuously differentiable function as $\text{QRR}_a$, where $a$ is the value of the parameter used for determining $E_i$ and $E_{i+1}$.

$$
q_{i-1}(x) = 12x^2 - 24x + 39; \quad q_{i-1}(0) = 39
$$

$$
q_i(x) = 60x^2 + 20x + 25; \quad q_i(0) = 25
$$

$$
q_{i+1}(x) = -600x^2 + 10x + 77.5; \quad q_{i+1}(0) = 77.5
$$

Figure 2.2: Illustration of imaging-consistent reconstruction/restoration algorithms. (a) QRsR $-.5$; (b) QRR $-.5$. Note that there is a half pixel phase shift between reconstruction and restoration.

An example of QRsR and QRR with $a = -.5$ is illustrated in Figure 2.2. The
size of the image is 7 pixels, with \( V_i = 30 \) and \( V_{i+1} = 70 \). Figure 2.2a shows QRsR \( = -5 \); Figure 2.2b QRR \( = -5 \). It is interesting to note that Figure 2.2a explains why QRsR provides a sharper image than bi-linear resampling. As can be easily seen from Figure 2.2a, to compute an intensity value located somewhere between \( p_i \) and \( p_{i+1} \), bi-linear resampling is constrained to values between the data, i.e., somewhere between 30 and 70. For QRsR, however, only the integral over the pixel and the end point values are constrained, allowing it to take on values below 30 or beyond 70 and thus allowing a sharper image. Also note that there is a half pixel phase shift between reconstruction and restoration.

### 2.3.2 A Quadratic Imaging-Consistent Algorithm Assuming a Gaussian PSF Filter

A second quadratic method is obtained by assuming that the PSF is given by \( h(x) \), a cubic B-spline approximation to a Gaussian. This PSF has four cubic segments in the single input pixel which are used to weigh the integral. Again, we are assuming centered pixels. To ensure that the function is in \( F_0 \), we need continuity, but we also desire a local method. Hence, we define the value of the reconstruction at the pixel boundaries \( k_i \) and \( k_{i+1} \), which we refer to as \( E_i \) and \( E_{i+1} \).

Given the value at the edges of the pixel, an additional constraint that the integral across the pixel—weighted by the cubic B-spline—must equal \( V_i \) gives exactly three constraints. They are

\[
p_i(-1/2) = E_i, \quad p_i(1/2) = E_{i+1}, \quad \int_{-1/2}^{1/2} p_i(x) h(x) \, dx = V_i
\]
where the cubic B-spline (after scaling) is defined by

\[
    h(x) = \begin{cases} 
        h_1(x) = \frac{(128x^3 + 192x^2 + 96x + 16)}{3}, & -1/2 \leq x \leq -1/4 \\
        h_2(x) = \frac{(-384x^3 - 192x^2 + 8)}{3}, & -1/4 \leq x \leq 0 \\
        h_3(x) = \frac{(384x^3 - 192x^2 + 8)}{3}, & 0 \leq x \leq 1/4 \\
        h_4(x) = \frac{(-128x^3 + 192x^2 - 96x + 16)}{3}, & 1/4 \leq x \leq 1/2 \\
        0, & 1/2 < |x| 
    \end{cases}
\]

From this, one can determine the quadratic polynomial to be

\[
    p_i(x) = \frac{1}{22} \left( 48(E_i + E_{i+1} - 2V_i)x^2 - 22(E_i - E_{i+1})x - (E_i + E_{i+1} - 24V_i) \right)
\]

where \(-1/2 \leq x \leq 1/2\). This then defines the quadratic restoration algorithm using the B-spline approximation to a Gaussian PSF and a free parameter \(a\), where \(a\) is the value of the parameter used for determining \(E_i\) and \(E_{i+1}\). We abbreviate this as QRsG \(a\).

The above methods are always defined, though at times they will produce reconstructions with negative values. We note that the value of the second derivative on a segment is simply

\[
    \frac{d^2 p_i(x)}{dx^2} = \frac{48}{11} (E_i + E_{i+1} - 2V_i) \\
    = \frac{1}{11} \left( 6aV_{i-2} + 24V_{i-1} - (48 + 12a)V_i + 24V_{i+1} + 6aV_{i+2} \right).
\]

In general, the maximum value of the second derivative is on the order of the maximum step change in image intensity values.

To define the reconstruction algorithm, we simply blur the resulting restoration by a cubic B-spline approximation to a Gaussian PSF of the same size. One can derive a functional form for this that results in four six-degree polynomials per segment connected as a continuously differentiable function. Let \(v = x - 1/2, z_1 = z - x, \text{ and } z_2 = z - x + 1\).
The four six-degree polynomials are

\[ P_i^{(1)}(x) = \int_{\nu}^{\nu+1/4} p_i(z)h_1(z_1) \, dz + \int_{\nu+1/4}^{\nu+2/4} p_i(z)h_2(z_1) \, dz + \int_{\nu+2/4}^{\nu+3/4} p_i(z)h_3(z_1) \, dz \]
\[ + \int_{\nu+3/4}^{1/2} p_i(z)h_4(z_1) \, dz + \int_{-1/2}^{\nu} p_{i+1}(z)h_4(z_2) \, dz \]
\[ = ((-4096V_{i+1} - 2048E_i + 2048E_{i+2} + 4096V_i)x^6 \]
\[ + (-17920E_{i+1} - 3328E_i - 3328E_{i+2} + 12288V_i + 12288V_{i+1})x^5 \]
\[ + (-5760V_i + 2880E_{i+1} + 2880E_i)x^4 \]
\[ + 1320(E_{i+1} - E_i)x + 1320V_i)/1320, \]

\[ P_i^{(2)}(x) = \int_{\nu}^{\nu+1/4} p_i(z)h_1(z_1) \, dz + \int_{\nu+1/4}^{\nu+2/4} p_i(z)h_2(z_1) \, dz + \int_{\nu+2/4}^{\nu+3/4} p_i(z)h_3(z_1) \, dz \]
\[ + \int_{\nu+3/4}^{1/2} p_{i+1}(z)h_3(z_2) \, dz + \int_{-1/2}^{\nu} p_{i+1}(z)h_4(z_2) \, dz \]
\[ = ((-12288V_i + 12288V_{i+1} + 6144E_i - 6144E_{i+2})x^6 \]
\[ + (-61440V_{i+1} + 53760E_{i+1} + 22272E_{i+2} - 12288V_i - 2304E_i)x^5 \]
\[ + (-89600E_{i+1} - 8960E_i - 24320E_{i+2} + 46080V_i + 76800V_{i+1})x^4 \]
\[ + (5760E_i + 44800E_{i+1} - 25600V_i + 10880E_{i+2} - 35840V_{i+1})x^3 \]
\[ + (1280E_i - 2560E_{i+2} - 8320E_{i+1} + 960V_i + 8640V_{i+1})x^2 \]
\[ + (308E_{i+2} - 1056V_{i+1} + 2720E_{i+1} - 864V_i - 1108E_i)x \]
\[ + (52V_{i+1} - 15E_{i+2} - 70E_{i+1} + 1364V_i - 11E_i))/1320, \]

\[ P_i^{(3)}(x) = \int_{\nu}^{\nu+1/4} p_i(z)h_1(z_1) \, dz + \int_{\nu+1/4}^{\nu+2/4} p_i(z)h_2(z_1) \, dz + \int_{\nu+2/4}^{\nu+3/4} p_i(z)h_3(z_1) \, dz \]
\[ + \int_{\nu+3/4}^{1/2} p_{i+1}(z)h_3(z_2) \, dz + \int_{-1/2}^{\nu} p_{i+1}(z)h_4(z_2) \, dz \]
\[ = ((12288V_i - 12288V_{i+1} - 6144E_i + 6144E_{i+2})x^6 \]
\[ + (86016V_{i+1} - 53760E_{i+1} - 34560E_{i+2} - 12288V_i + 14592E_i)x^5 \]
\[ + (179200E_{i+1} - 5120E_i + 71680E_{i+2} - 46080V_i - 199680V_{i+1})x^4 \]
where \( P_i^{(1)}(x) \) is defined on the interval \( 0 \leq x \leq 1/4 \); \( P_i^{(2)}(x) \) on the interval \( 1/4 \leq x \leq 1/2 \); \( P_i^{(3)}(x) \) on the interval \( 1/2 \leq x \leq 3/4 \); \( P_i^{(4)}(x) \) on the interval \( 3/4 \leq x \leq 1 \). We refer to the resulting (globally) continuously differentiable function as QRG \( a \), where \( a \) is the value of the parameter used for determining \( E_i \) and \( E_{i+1} \).

### 2.3.3 An Imaging-Consistent Algorithm with 2-Piece Cubics Assuming a Rect PSF Filter

Additional smoothness, in terms of global differentiability, seems to be a natural extension. We first tried a single quartic polynomial per segment, but this resulted in too much
ringing. Hence, we decided to try two cubic polynomials per segment joined such that the overall function was continuously differentiable. Again, to ensure continuity but still allow the method to be local, we need to define the value of the reconstruction at the pixel boundaries \( k_i \) and \( k_{i+1} \), which we will refer to as \( E_i \) and \( E_{i+1} \). In addition, we need derivative values, which we can obtain from cubic convolution. We refer to the left and right side derivative values as \( E_i \) and \( E_{i+1} \), respectively. With the integral constraint, this totals seven constraints:

\[
\begin{align*}
& r_i^{(1)}(-1/2) = E_i, \quad r_i^{(2)}(1/2) = E_{i+1}, \quad r_i^{(1)}(0) - r_i^{(2)}(0) = 0, \\
& r_i^{(1)'}(x) = E_i', \quad r_i^{(2)'}(x) = E_{i+1}', \quad r_i^{(1)'}(0) - r_i^{(2)'}(0) = 0, \\
& \int_{-1/2}^{0} r_i^{(1)}(z) \, dz + \int_{0}^{1/2} r_i^{(2)}(z) \, dz = V_i.
\end{align*}
\]

But we have eight parameters. We define the method with a general free parameter \( K \), but any value other than zero introduces significant frequency-sample ripple.

Solving the seven equations mentioned above, we have

\[
\begin{align*}
& r_i^{(1)}(x) = \left( (576E_i + 112E_i' - 768V_i + 192E_{i+1} - 16E_{i+1}' + 96K)/24 \right) x^3 \\
& \quad + \left( (432E_i + 60E_i' - 576V_i + 144E_{i+1} - 12E_{i+1}' + 96K)/24 \right) x^2 \\
& \quad + Kx - (12E_i + E_i' - 48V_i + 12E_{i+1} - E_{i+1}')/24, \\
& r_i^{(2)}(x) = \left( (-192E_i - 16E_i' + 768V_i - 576E_{i+1} + 112E_{i+1}' + 96K)/24 \right) x^3 \\
& \quad + \left( (144E_i + 12E_i' - 576V_i + 432E_{i+1} - 60E_{i+1}' - 96K)/24 \right) x^2 \\
& \quad + Kx - (12E_i + E_i' - 48V_i + 12E_{i+1} - E_{i+1}')/24.
\end{align*}
\]

where \( r_i^{(1)} \) is defined on the interval \([-1/2, 0]\); \( r_i^{(2)} \) on the interval \([0, 1/2]\). The integral of the pair of segments with the knot at \((k_i + k_{i+1})/2\) over the interval \( k_i \) to \( k_{i+1} \) is exactly
$V_i$. Using cubic convolution to derive $E'_i$ and $E'_{i+1}$ (See Section 2.4.1 for details.), we have

$$E'_i = \frac{1}{4} (-a V_{i-2} - (a + 6)V_{i-1} + (a + 6)V_i + a V_{i+1}),$$

$$E'_{i+1} = \frac{1}{4} (-a V_{i-1} - (a + 6)V_i + (a + 6)V_{i+1} + a V_{i+2}).$$

To define the reconstruction algorithm, we simply blur the resulting restoration by a Rect filter of the same size. One can derive a functional form for this that results in two quartic polynomials connected as a continuously differentiable function. Symbolically, they are

$$R^{(1)}_i(x) = \int_{x-1/2}^{0} r^{(1)}_i(z) \, dz + \int_{0}^{1/2} r^{(2)}_i(z) \, dz + \int_{-1/2}^{x-1/2} r^{(1)}_{i+1}(z) \, dz$$

$$= ((48 V_i + 12 E'_{i+2} - E'_i + 48V_{i+1} - 36E_i + 8E'_{i+1} - 7E'_i + 24E_{i+1})x^4$$

$$+ (-48 V_i - 12 E_{i+2} + E'_{i+2} + 48V_{i+1} + 36E_i - 10E'_{i+1} + 9E'_i$$

$$- 24E_{i+1})x^3$$

$$+ (3E'_{i+1} - 3E'_i)x^2 + 6(-E_i + E_{i+1})x + 6V_i)/6,$$

$$R^{(2)}_i(x) = \int_{x-1/2}^{1/2} r^{(2)}_i(z) \, dz + \int_{0}^{1/2} r^{(1)}_{i+1}(z) \, dz + \int_{-1/2}^{x-1/2} r^{(2)}_{i+1}(z) \, dz$$

$$= ((-36 E_{i+2} + 7E'_i + 48V_{i+1} + 24E_{i+1} + E'_i + 12E_i - 8E'_{i+1} - 48V_i)x^4$$

$$+ (-144 V_{i+1} - 19E'_{i+2} - 72E_{i+1} + 108E_{i+2} - 36E_i + 144V_i - 3E'_i$$

$$+ 22E'_{i+1})x^3$$

$$+ (-21 E'_{i+1} + 3E'_i - 108 E_{i+2} + 36E_i + 72E_{i+1} - 144V_i + 144V_{i+1}$$

$$+ 18E'_{i+2})x^2$$

$$+ (-E'_i - 30E_{i+1} + 48V_i - 12E_i + 42E_{i+2} - 48V_{i+1} + 8E'_{i+1} - 7E'_i + 24E_{i+1})x$$

$$+ (-6E_{i+2} + E'_{i+2} + 6V_{i+1} - E'_{i+1} + 6E_{i+1}))/6$$

where $R^{(1)}_i$ is defined on the interval $0 \leq x \leq 1/2$; $R^{(2)}_i$ on the interval $1/2 \leq x \leq 1$. We
refer to the resulting reconstruction algorithms as TPCRR \( a \), where \( a \) is the value of the parameter used for determining \( E_i, E_{i+1}, E'_i, \) and \( E'_{i+1} \).

## 2.4 Proofs

In this section, we begin by deriving \( E_i \) and its derivative \( E'_i \) in Section 2.4.1 and then prove that if linear interpolation is used to determine \( E_i \), the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of \( a = -0.5 \) is provided in Section 2.4.2.

### 2.4.1 Derivation of \( E_i \) and \( E'_i \)

Cubic convolution is a third-degree interpolation algorithm with its interpolation kernel defined by

\[
h(x) = \begin{cases} 
  (a + 2)|x|^3 - (a + 3)|x|^2 + 1, & 0 \leq |x| < 1 \\
  a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & 1 \leq |x| < 2 \\
  0, & 2 \leq |x|
\end{cases}
\]

For convenience, the cubic convolution kernel \( h \) can be put in the form

\[
h(x) = \begin{cases} 
  h_1(x) = -a(x - 2)^3 - 5a(x - 2)^2 - 8a(x - 2) - 4a, & 0 \leq x \leq 1 \\
  h_2(x) = -(a + 2)(x - 1)^3 - (a + 3)(x - 1)^2 + 1, & 0 \leq x \leq 1 \\
  h_3(x) = (a + 2)x^3 - (a + 3)x^2 + 1, & 0 \leq x \leq 1 \\
  h_4(x) = a(x + 1)^3 - 5a(x + 1)^2 + 8a(x + 1) - 4a, & 0 \leq x \leq 1 \\
  0, & 1 < |x|
\end{cases}
\]

Clearly, this does not change the meaning of the cubic convolution kernel \( h \) except that it is easier to deal with because we no longer need to worry about at which of the four subintervals the value of \( x \) is located and which of the four functions has to be used.
Figure 2.3: Derivation of $E_i$ and $E'_i$ assuming $a = -.5$.

Using cubic convolution (See Figure 2.3), one can derive a functional form for reconstruction that results in a cubic polynomial that spans from the center of one input pixel to the next. The cubic polynomial that spans from the center of pixel $i-1$ to that of pixel $i$ is expressed by

$$C_{i-1}(x) = V_{i+1}h_1(x) + V_ih_2(x) + V_{i-1}h_3(x) + V_{i-2}h_4(x)$$

$$= (aV_{i-2} + (a + 2)V_{i-1} - (a + 2)V_i - aV_{i+1})x^3$$

$$+ (-2aV_{i-2} - (a + 3)V_{i-1} + (2a + 3)V_i + aV_{i+1})x^2$$

$$+ (aV_{i-2} - aV_i)x + V_{i-1}$$

where $0 \leq x \leq 1$.

Letting $x = 1/2$, we get

$$E_i = \frac{1}{8} (aV_{i-2} + (4 - a)V_{i-1} + (4 - a)V_i + aV_{i+1})$$

Then, taking the first derivative of $C_{i-1}(x)$ with respect to $x$ and letting $x = 1/2$, we get

$$E'_i = \frac{1}{4} (-aV_{i-2} - (a + 6)V_{i-1} + (a + 6)V_i + aV_{i+1})$$
This gives $E_i$ and its derivative $E'_i$ for all $i$.

2.4.2 The Relationship between QRR and Cubic Convolution

We now proceed to prove that if linear interpolation is used to determine $E_i$, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of $a = -.5$. This is exactly the same as saying that $a = -1/2$ and $b = 1/2$, where $b$ is the parameter of linear interpolation.

As shown in Section 2.3.1, for pixels $i-1$ and $i$, the restored functions are given by

\begin{align*}
q_{i-1}(x) &= 3(E_{i-1} + E_i - 2V_{i-1})x^2 - (E_{i-1} - E_i)x - (E_{i-1} + E_i - 6V_{i-1})/4 \\
q_i(x) &= 3(E_i + E_{i+1} - 2V_i)x^2 - (E_i - E_{i+1})x - (E_i + E_{i+1} - 6V_i)/4
\end{align*}

One can derive a functional form for reconstruction that results in one cubic polynomial that spans from the center of one input pixel to the next. The cubic polynomial that spans from the center of pixel $i-1$ to that of pixel $i$ is expressed by

\begin{align*}
Q_{i-1}(x) &= \int_{x-1/2}^{x+1/2} q_{i-1}(z) dz + \int_{x-1/2}^{x+1/2} q_i(z) dz \\
&= (E_{i+1} - E_{i-1}) - 2(V_i - V_{i-1})) x^3 \\
&\quad + (2E_{i-1} - E_i - E_{i+1} + 3(V_i - V_{i-1})) x^2 + (E_i - E_{i-1}) x + V_{i-1}
\end{align*}

where $0 \leq x \leq 1$ (or $p_i \leq t \leq p_{i+1}$).

By assumption, letting $E_{i-1} = (1 - b)V_{i-2} + bV_{i-1}$, $E_i = (1 - b)V_{i-1} + bV_i$, and $E_{i+1} = (1 - b)V_i + bV_{i+1}$, we have

\begin{align*}
Q_{i-1}(x) &= ((-1 + b)V_{i-2} + (-b + 2)V_{i-1} + (-1 - b)V_i + bV_{i+1}) x^3 \\
&\quad + ((2 - 2b)V_{i-2} + (-4 + 3b)V_{i-1} + 2V_i - bV_{i+1}) x^2
\end{align*}
one and only one polynomial of degree

Subtracting \( Q_{i-1}(x) \) from \( C_{i-1}(x) \), we have

\[
\begin{align*}
& ((V_{i-1} - V_i - V_{i+1} + V_{i-2}) a + (-V_{i-2} + V_i - V_{i+1} + V_{i-1}) b + V_{i-2} - V_i) x^3 \\
+ & ((2V_i + V_{i+1} - 2V_{i-2} - V_{i-1}) a + (2V_{i-2} - 3V_{i-1} + V_{i+1}) b - 2V_{i-2} + V_{i-1} + V_i) x^2 \\
+ & ((V_{i-2} - V_i) a + (-V_{i-2} + 2V_{i-1} - V_i) b + V_{i-2} - V_{i-1}) x
\end{align*}
\]

Clearly, in order to make \( C_{i-1}(x) = Q_{i-1}(x) \) for all \( x \), the coefficients must be all equal to zero. Thus, equating them to zero and solving the system of linear equations

\[
\begin{bmatrix}
V_{i-1} - V_i - V_{i+1} + V_{i-2} & -V_{i-2} + V_i - V_{i+1} + V_{i-1} \\
2V_i + V_{i+1} - 2V_{i-2} - V_{i-1} & 2V_{i-2} - 3V_{i-1} + V_{i+1} \\
V_{i-2} - V_i & -V_{i-2} + 2V_{i-1} - V_i
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
= 
\begin{bmatrix}
-V_{i-2} + V_i \\
2V_{i-2} - V_{i-1} - V_i \\
-V_{i-2} + V_{i-1}
\end{bmatrix}
\]

we have \( a = -1/2 \) and \( b = 1/2 \). This completes the proof.

Mathematically, this is easy to explain. Using cubic convolution to derive \( E_i \), \( Q_{i-1}(x) \) has six points of support \( V_k, k = i - 3, \ldots, i + 2 \); however, \( C_{i-1}(x) \) has only four points of support \( V_k, k = i - 2, \ldots, i + 1 \). Using linear interpolation, both \( C_{i-1}(x) \) and \( Q_{i-1}(x) \) have exactly the same four points of support. The fundamental result of algebra says that if \( C_{i-1}(x) \) and \( Q_{i-1}(x) \) are two polynomials of degree \( \leq 3 \) which agree at the four distinct points \( V_k, k = i - 2, \ldots, i + 1 \), then \( C_{i-1}(x) = Q_{i-1}(x) \) identically [Conte and Boor-1980]. This is exactly the same as saying that there exists one and only one polynomial of degree \( \leq 3 \) which interpolates the four distinct points \( V_k, k = i - 2, \ldots, i + 1 \). Thus, \( C_{i-1}(x) \) and \( Q_{i-1}(x) \) must be the same function.

### 2.5 Experimental Evaluation Using Real Images

In this section, we present a short experimental comparison of the different algorithms using real images. Readers are referred to [Boult and Wolberg-1993, Figure 5] for details.
We point out that the imaging-consistent algorithms all assume that the data are centered within the pixel, while the cubic convolution, linear interpolation, and cubic spline methods all assume the data are at the left of the pixel. Thus, there is a four-pixel shift in the imaging-consistent constructions. This could, of course, be corrected if one prefers the left-justified view of the data.

The restorations have significantly more ringing, as they amplify frequencies near the sampling rate. The fact that the restorations do not look too much like what we expect of the original signal suggests that the restoration model (the PSF and/or the functional model) is rather weak. Our algorithm only allowed discontinuities of the second derivative at boundaries of the original samples. To get a better restoration, this needs to be relaxed to allow them to occur at any point with the original pixel.

### 2.6 General Error Properties of Imaging-Consistent Algorithms

The imaging-consistent reconstruction follows quite naturally from a general approach to algorithm development known as information-based complexity (IBC) (See [Traub et al.-1988]). Because of the relationship, we know that the imaging-consistent algorithms enjoy good error properties for many definitions of error.

In the IBC theory, there are a few key concepts which apply to our situation. First, the only available information is mathematically captured by the *information operator*, which converts an unknown function to input values. In our case, the information operator is area sampling process. Second, this operator is applied to an unknown function from an *a priori* known class of functions, which in our case is $F_0$ or $F_1$ defined previ-
ously. Third, error is measured in another space, e.g., the sum of squares of pointwise distance between functions. Finally, the concept of an interpolatory algorithm is defined to be one which produces a function from the class of allowable functions such that when the information operator is applied to this function, it returns the measured data. That is, the function interpolates (in a generalized sense) the measured data.

Most importantly, it has been proven that such interpolatory algorithms enjoy strong error properties. In particular, interpolatory algorithms have an error at most twice that of any algorithm for any error measure defined as a weighted norm on the space of solutions (e.g., $L^2$, or even a weighted least-square measure). Note that most image-quality measures that result in a single number are error measures of this type, e.g., the QSF measure of [Drago and Granger-1985, Granger-1974], as well as the extensions of this which include contrast effects or any other weighted integral of the MTF. The measure of [Park and Schowengerdt-1982], when weighted and integrated over frequency $v$, is also this type of error measure. Thus, an IBC-type interpolatory algorithm is, within a factor of 2, an optimal algorithm for each of these image-reconstruction quality measures. Of course, being optimal does not make them any good because the error may be infinite or very large. If, however, a good algorithm exists according to the assumed mathematical model (function class, information operator, and error measure), then the interpolatory algorithm should be acceptable. In our examples, if there are enough samples in the image so that variation of the second derivative is small, then error of the optimal algorithm is small and the imaging-consistent algorithms should perform well.

By definition, an imaging-consistent reconstruction algorithm is an (IBC) interpolatory algorithm if the reconstructed function is from the class (in our case, that means ensuring that the second derivative is below the prescribed bound). In fact, the imaging-
consistent property is almost analogous to the IBC concept of interpolatory algorithm.
We defined the term “imaging-consistent algorithm” to avoid confusion with the more
traditional use, in image processing, of the term interpolatory algorithm.

If the data had been taken as point samples, then an interpolatory algorithm would
simply be one that interpolated the data in the traditional sense and had properly bounded
second derivative. This is probably one reason why cubic splines are so effective: they
interpolate the data and minimize a measure based on the second derivative. If we assume
that data were generated by area sampling, an interpolatory algorithm must be one such
that when the resulting function is integrated (weighted by the PSF) the results interpolate
the data. Our algorithms satisfy this property.

While we started this whole endeavor assuming that area sampling was important,
we knew that tremendous progress has been made assuming (properly filtered) point sam-
pies. We also knew that existing algorithms performed reasonably well when measured
subjectively. To the best of our knowledge, all existing quantitative measures of image
reconstruction quality are based on spectral analysis (with the implicit assumption of
point sampling) and measure performance with respect to the “ideal reconstruction,” a
sinc function. Thus, we were faced with the question of how to access the performance
of these new filters. Rather than pursuing the arduous task of psychological testing of
subjective performance, we decided to use traditional spectral analysis. We felt that we
would be happy if we came close to traditional filters like cubic convolution. As was
shown in [Boult and Wolberg-1993], the new imaging-consistent reconstruction algo-
rithms generally not only outperformed cubic convolution, but they generally also rivaled
cubic splines.
2.7 Conclusion

This chapter introduced a new class of reconstruction algorithms that are fundamentally different from traditional approaches. We treat image values as area samples generated by non-overlapping integrators, which is consistent with the image formation process in CCD and CID cameras. We obtained superior results by formulating reconstruction as a two-stage process: image restoration followed by application of the point spread function of the imaging sensor. Efficient local techniques for image restoration are derived to invert the effects of the PSF and estimate the underlying image that passed through the sensor. We define the imaging-consistency constraint which requires the approximate reconstruction to be the exact reconstruction for some function in the allowable class of functions. This class of functions is defined by bounding the maximum absolute value of the second derivative of the underlying continuous signal. The error of the new algorithms can be shown to be at most twice the error of the optimal algorithm for a wide range of optimality constraints.

The reconstruction algorithms derived herein are local methods that compare favorably to cubic convolution, a well-known local technique, and they even rival global algorithms such as interpolating cubic splines. Evaluations (See [Boult and Wolberg-1993]) were made by comparing their passband and stopband performances in the frequency domain, as well as by direct inspection of the resulting images in the spatial domain. We proved that if linear interpolation is used to derive the value of the reconstruction at the pixel boundaries, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of $a = -\frac{5}{6}$.

The results of this work will be applied to the general problem of nonlinear image warping and reconstruction from nonuniform samples. In addition, we will extend our
technique to deal with other point spread functions, including those that overlap.
Chapter 3

Imaging-Consistent Warping

This chapter introduces integrating resamplers as an efficient method for image warping using the class of imaging-consistent reconstruction/restoration algorithms described in Chapter 2. Examples are given to show the use of the integrating resamplers described herein for two low-level computer vision applications. The first is geometric correction for images degraded by radial lens distortion, demonstrated as a preprocessing step in correlation-based stereo matching. The second is warping-based data fusion for polarization computations in a small-baseline multi-camera system. These two applications are chosen because they exemplify the kinds of applications for which the integrating resamplers and the imaging-consistent reconstruction/restoration algorithms are designed—problems that use warped intensity values (as opposed to edge structure).

3.1 Introduction

Image warping has been around almost as long as image processing, allowing users to reshape the image geometry. Image warping requires the underlying image to be “resam-
pled” at non-integer locations; hence, it requires image reconstruction. When the goal of warping is to produce output for human viewing, only moderately accurate image intensities are needed. In these cases, techniques using bi-linear interpolation have been found sufficient. However, as a preprocessing step for vision, the precision of the warped intensity values is often important. For these problems, bi-linear image reconstruction may not be sufficient.

This chapter shows how to efficiently warp images using an image reconstruction technique that includes a simple sensor model. By coupling the degradation model of the imaging system directly into the reconstruction process, we can derive better reconstruction techniques which more accurately mimic a “digital lens.” This realization leads to the idea of imaging-consistent reconstruction/restoration algorithms proposed in Chapter 2.

This chapter is organized as follows. Section 3.2 presents the integrating resampler, an efficient method for warping using imaging-consistent reconstruction/restoration algorithms. This algorithm is well suited for today’s pipelined micro-processors. In addition, the integrating resampler can allow for modifications of the intensity values to better approximate the warping characteristics of real lenses. Section 3.3 demonstrates these ideas on two low-level computer vision applications where we compare the integrating resampler with bi-linear resampling. Conclusion is given in Section 3.4.

3.2 Integrating Resampler

To define an imaging-consistent warping algorithm (also known as the integrating resampler), we generalize the idea of the imaging-consistent reconstruction/restoration
algorithms. Whereas imaging-consistent reconstruction algorithms simply assume the degradation models are identical for both input and output; imaging-consistent warping algorithms go one step further, allowing

1. both input and output to have their own degradation model, and

2. the degradation model to vary its size for each output pixel.

The imaging-consistent algorithms in Chapter 2 are linear filters. For linear and affine spatial transformations, the traditional implementation would be in terms of convolution with the impulse response. However, we presented reconstruction/restoration in functional form because we designed them for use in what we call the integrating resampling approach. This section describes integrating resamplers and their use in image warping.

As described before, our model of image formation requires the image to be spatially sampled with a finite area sampler. This is tantamount to a weighed integral being computed on the input function. Because we have a functional form for the reconstruction/restoration, we can simply integrate this function with the PSF for the output area sampler. In this section, we assume the output sampler has a Rect PSF, though there is no limitation on the degradation models that one can use.

When we are resampling the image and warping its geometry in a nonlinear manner, this new approach allows us to efficiently do both pre-filtering and post-filtering. Because we have already determined a functional form for the input, no spatially-varying filtering is needed, as would be the case if direct inverse mapping were done.

Computing the exact value of the restored function weighted by the PSF could be done in functional form if the mapping function has a functional inverse and the PSF is
simple (as in this example). In general, however, it cannot be done in closed form, and numerical integration is required. To reduce the computational complexity, we propose a scheme where for each input pixel, we use a linear approximation to the spatial warping within that pixel, but use the full non-linear warp to determine the location of pixel boundaries. This integrating resampler, presented in Figure 3.2, also handles antialiasing of partial pixels in a straightforward manner.

Assume \( n \) input pixels are being mapped into \( k \) output pixels according to the mapping function \( m(t) \). Let \( m_i \) be the mapped location of pixel \( i \), for \( i = 0, \ldots, n \). Compute \( q_j \), \( j = 0, \ldots, k \), as the linear approximation to the location of \( m^{-1}(j) \), as shown in Figure 3.1. Note that we are assuming in Figure 3.1 that the mapping function is strictly increasing to avoid fold-over problems. See [Wolberg and Boult-1989] for an approach to modeling fold-over.

\[
\text{for } (i = j = 0; j \leq k; j++) \{ \\
\quad \text{while } (i < n - 1 \&\& m_{i+1} < j) \ i++; \\
\quad q_j = i + (j - m_i)/(m_{i+1} - m_i); \\
\}\]

Figure 3.1: Linear approximation to the location of \( m^{-1}(j) \).

In order to allow efficient computation of the integral as well as to perform proper antialiasing, the algorithm—given as pseudo code in Figure 3.2 and referred to as QRW in the following chapters—runs along the input and output determining in which image it will next cross a pixel boundary. To do this, we have two variables: \( \text{inseg} \in [0,1] \), which represents the fraction of the current input pixel left to be consumed, and \( \text{outseg} \), which specifies the amount of input pixel(s) required to fill the current output pixel. If we assume that the output PSF is a Rect filter, the closed form definite integral \( R(t) \) of
the restoration \( r(t) \) from point \( a \) to \( b \) can be derived similar to Eq. (2.3). Other imaging-consistent algorithms yield different forms of \( R \).

Whenever \( \text{inseg} < \text{outseg} \), we know that the input pixel will finish first, so we can consume it and update our state. If, on the other hand, it happens that \( \text{inseg} \geq \text{outseg} \), the output pixel will finish first, so we produce an output and update our state. Thus, we process in a loop such that each time, we either consume one input pixel or produce one output pixel; the algorithm requires at most \( k + n \) iterations.

The underlying idea of this integrating resampler can be found in the work of Fant [Fant-1986] who proposed a similar algorithm for the special case of linear reconstruction. Fant’s original work can be viewed as imaging-consistent with a piecewise constant image resulting in a linear form for \( R(t) \). Our contribution is twofold:

1. the generalization to deal with more advanced reconstruction algorithms, and

2. providing for a modeling of real lens effects by using a modeling of real warps that affects the image radiance.

In Fant’s original work, the goal was to warp images for graphic (or visual) effects, and hence to affect geometry without disturbing the intensities. To do this, the algorithm maintained knowledge of the input size and normalized the integral to account for this size, giving a normalized intensity. Thus, if a constant image was stretched to twice its normal width, it would just change shape but retain the same intensities. Since we may be interested in the modeling of lens characteristics, we realized that we do not always want to do this normalization. If a lens was placed into an imaging system so as to double the width of the image on the sensor plane, then the value measured would be halved.

If an input scene \( f \) is warped according to some mapping function \( m(t) \), then the measured intensity at location \( O_j \) will depend only on the energy from \( f \) that is warped
Pad the input; compute $k_l$, $k_r$, and $i_l$, the indices to the leftmost and rightmost output pixels and the index to the leftmost input pixel that contributes to the output; and compute the linear approximation to the location of $m^{-1}(j)$, for $j = k_l, \ldots, k_r + 1$.

normalizingfactor = $q_k + 1 - q_{k_l}$; // set up for normalization
$q_k = \text{MAX}(q_k, 0)$; // ensure that $q_k$ is nonnegative
inseg = 1.0 - FRACTION($q_k$); // fraction of input pixel left to be consumed
outseg = $q_k + 1 - q_{k_l}$; // #input pixels mapped onto one output pixel
acc = 0.0; // reset accumulator for next output pixel

for ($j = 0; j < q_k; j++$) out[$j++] = 0; // zero out the garbage at left end
for ($i = i_l; j = k_l; j < = k_r; )$ { // while there is output to produce
    // Use the current pixel (in[$i$]) and its neighbors to update $R()$, the integral of the restoration $r()$.  
    leftpos = 1.0 - inseg; // get left endpoint for integration
    if (inseg < outseg) { // if we will consume input pixel first
        acc += $R(1) - R($leftpos$)$; // add integral to end of output pixel
        $i++$; // index into next input pixel
        if ($i == n$) { // check end condition
            if (normalize) acc /= normalizingfactor; // normalize the output, if appropriate
            out[$j$] = acc; // init output
            break; // exit from the loop
        }
    }
    outseg -= inseg; // inseg portion has been filled
    inseg = 1.0; // new input pixel will be available
}
else { // Else we will produce output pixel first
    acc += $R($leftpos + outseg$) - R($leftpos$)$; // add integral to end of output pixel
    if (normalize) acc /= normalizingfactor; // normalize the output, if appropriate
    out[$j$] = acc; // init output
    $j++$; // index into next output pixel
    acc = 0.0; // reset accumulator for next output pixel
    inseg -= outseg; // outseg portion of input has been used
    outseg = $q_{j+1} - q_j$; // new output size
    normalizingfactor = outseg; // need for normalization
}

for ($j = k_r + 1; j < k; j++$) out[$j++] = 0; // zero out the garbage at right end

Figure 3.2: The integrating resampler assuming a Rect PSF filter. See text for discussion.
according to $m$ to reach the pixel. This assumes no blurring effects while in reality, whatever caused the geometric distortion would likely have also resulted in some blurring. It also presumes that the sensor is linear with respect to its input, which may not be the case near the extremes of the camera’s dynamic range, or in cameras with non-unity gamma (gain), automatic gain control, or auto irises. Still, it is a better approximation than just resampling which assumes that the intensity is independent of the geometric distortions.

To make it easier to understand the idea of the integrating resampler described above, a one-dimensional example is given in Figure 3.3. This example demonstrates that 4 input pixels are being mapped to 5 output pixels according to the mapping function $m(t)$. Consider the arrays shown in Figure 3.3. The first array specifies the values of $m_i$ for $i = 0, \ldots, 4$. These values represent new coordinates for their respective input pixels. For instance, the leftmost input pixel produces the output in the range $[0.50, 2.50]$. The next input pixel produces the output in the range $[2.50, 2.75]$. This continues until the last input pixel is consumed, producing the output in the range $[3.25, 4.50]$. Note that this requires the first array to have an additional element to define the contribution of the last input pixel to the output.

The second array specifies the distribution range that each input pixel assumes in the output. It is simply the difference between adjacent coordinates given in the first array. Large values represent stretching; small values correspond to compression.

The input intensity values are given in the third array. Their contributions to the output are marked by the dotted lines connecting the input to the output.

The last array specifies the values of $a_j$ for $j = 0, \ldots, 5$. These values are the linear approximation to the locations of $m^{-1}(j)$ for $j = 0, \ldots, 5$, computed using the
Figure 3.3: Integrating resampling example assuming QRR -.5.

linear approximation algorithm shown in Figure 3.1, as follows:

\[
q_0 = 0 + (0 - m_0)/(m_1 - m_0) = 0 + (0 - 0.50)/(2.50 - 0.50) = -0.25 \\
q_1 = 0 + (1 - m_0)/(m_1 - m_0) = 0 + (1 - 0.50)/(2.50 - 0.50) = 0.25 \\
q_2 = 0 + (2 - m_0)/(m_1 - m_0) = 0 + (2 - 0.50)/(2.50 - 0.50) = 0.75 \\
q_3 = 2 + (3 - m_2)/(m_3 - m_2) = 2 + (3 - 2.75)/(3.25 - 2.75) = 2.50 \\
q_4 = 3 + (4 - m_3)/(m_4 - m_3) = 3 + (4 - 3.25)/(4.50 - 3.25) = 3.60 \\
q_5 = 3 + (5 - m_3)/(m_4 - m_3) = 3 + (5 - 3.25)/(4.50 - 3.25) = 4.40
\]

They represent the range of input that each output pixel consumes. For instance, the leftmost output pixel consumes the input in the range \([-0.25, 0.25]\). The second output
pixel consumes the input in the range \([0.25, 0.75]\). This continues until the last output pixel is produced, consuming the input in the range \([3.60, 4.40]\). Note that for this example, partial pixels occur at both ends of the input; no intensity values are defined in the range \([-0.25, 0]\) and \([4, 4.40]\). The integrating resampler, presented in Figure 3.2, handles antialiasing of partial pixels in a straightforward manner, by assuming that the intensity values outside the range of input are zero, and the integration of zero weighted by a PSF is simply zero.

The normalizing factors are given in the sixth array. They are simply the difference between adjacent coordinates given in the last array. Note that this requires the last array to have an additional element to define the range of input that the last output pixel consumes.

The “normalized” output intensity values are given in the fifth array; the “unnormalized” output intensity values are shown in the fourth array. The range of input that each output pixel consumes is denoted by the solid lines connecting the output to the input. The normalized output is simply the unnormalized output divided by their respective normalizing factors given in the sixth array.

Table 3.1 gives step-by-step results of running the integrating resampler, presented in Figure 3.2, on the input given in Figure 3.3. The numbers shown on top of the table as well as in the first row (the row with \(i = -\)) are described in Figure 3.2. They represent the initial state of the integrating resampler.

The input pixel number \(i\) is given in the first column. Repeated input pixel number indicates the number of loops required to consume that input pixel. For instance, three loops, corresponding to the three rows with \(i = 0\), are required to consume that input pixel.
Table 3.1: Step-by-step results of running the integrating resampler shown in Figure 3.2 on the input shown in Figure 3.3.

The output pixel number \( j \) is given in the second column. Repeated output pixel number indicates the number of loops required to produce that output pixel. For instance, three loops, corresponding to the three rows with \( j = 2 \), are required to produce that output pixel.

The normalized output intensity values are shown in the third column; the unnormalized output intensity values are given in the fourth column. The normalized output are simply the unnormalized output divided by their respective normalizing factors given in the seventh column.

The fifth and sixth columns (i.e., the columns denoted \( R_r \) and \( R_l \) in Table 3.1) show, respectively, the values of \( R(\text{leftpos} + \text{outseg}) \) and \( R(\text{leftpos}) \). The integral of \( r \) from \( \text{leftpos} \) to \( \text{leftpos} + \text{outseg} \) is simply the difference between these two values, i.e., \( R(\text{leftpos} + \text{outseg}) - R(\text{leftpos}) \). Since we have a functional form for \( R \), the computation of \( R(\text{leftpos} + \text{outseg}) \) and \( R(\text{leftpos}) \) are straightforward.
The last two columns show the state of \textit{inseg} and \textit{outseg}, being updated and ready for producing next output pixel. For instance, the second row (the row with \( j = 0 \)) shows that \( \text{inseg} = 0.75 \) and \( \text{outseg} = 0.50 \) right after the leftmost output pixel \( O_0 \) was produced and before the integrating resampler is about to start producing \( O_1 \). For this example, partial pixels occur at both ends of the input. This can be seen from the first and last rows in Table 3.1. The first row shows that initially, \( \text{outseg} = 0.25 \) but \textit{normalizingfactor} = 0.50. This reflects the range of input, \([0, 0.25]\) instead of \([-0.25, 0.25]\), that the leftmost output pixel consumes. The last row shows that \( \text{inseg} = 0.00 \) but \( \text{outseg} = 0.40 \). This states that the entire input stream has been consumed, but the last output pixel is only partially produced. The statement \textbf{if} \((i == n) \{ \cdot \cdot \cdot \} \) in Figure 3.2 handles this end condition and produces the correct output, 35.97 unnormalized and 44.96 normalized.

3.3 Examples and Applications

We demonstrate these ideas on two low-level computer vision applications. See also [Boult and Wolberg-1992]. The first is correlation-based stereo with a pre-processing warp to enhance registration and correct for radial lens distortion. The second is warping-based data fusion for polarization computations in a small-baseline multi-camera systems.

3.3.1 Stereo Matching after Lens Distortion Correction

When doing stereo matching with correlation-based matchers, it is usually assumed that the difference between images is a simple translation. Pre-warping the image to remove distortions can significantly improve the matching speed by allowing epi-polar
constraints to result in a 1-D search. This idea is not new; however, previous work has
presumed that bi-linear interpolation was sufficient for the warping. In this section, we
briefly show how the integrating resampler can improve the quality of the matching re-
sults.

The test data used an inexpensive Sun video camera which has noticeable radial
distortion. Calibration and test images were from two converging views. The test images
are shown in Figure 3.4. Figure 3.4a and Figure 3.4b show the stereo pair before the
radial lens distortions are removed; Figure 3.4c and Figure 3.4d the stereo pair after the
radial lens distortions are removed.

Geometric correction refers to the process of recovering undistorted images from
distorted images. Though there exist several kinds of distortions, only radial lens dis-
tortion is considered in this example. The parameters that characterize the underlying
camera model are computed using the two-stage camera calibration method described in
[Tsai-1986, Tsai-1987, Chiang and Boult-1995] and Appendix A.

Given the camera parameters, the input images were warped to remove the lens
distortions. Two different warpings were applied, one using bi-linear warping (resam-
pling), and the other based on the integrating resampler. These “corrected” images were
then used as input to a sum-of-square distance stereo matching algorithm (7x7 template
window). To reduce any dependency on calibration error, a 2D search window was used.
The result of the matching is a dense disparity surface. In this demonstration, the test
images were from a planar object, and hence we evaluate the quality of warping by con-
sidering the best plane fit to the disparity data. For the images warped with the integrating
resampler, we found an average absolute error of 3.48 and an RMS error of 5.07. Us-
ing bi-linear interpolation, the average absolute error was 4.0, and the RMS error was
Figure 3.4: Images for stereo example. (a) left image; (b) right image; (c) (a) with radial lens distortions removed; (d) (b) with radial lens distortions removed.

5.62. The integrating resampler showed a 10-20\% improvement over pre-warping with bi-linear interpolation. This appears to be a sufficient gain to warrant the slightly larger
computational cost.

The sub-pixel accuracy of matching was achieved by fitting a quadratic (i.e., \( ax^2 + bx + c \)) through the point with a unique match and its two neighbors. If the match is not unique or if the match is unique but has only one neighbor, then that point is not used. Also, heuristics (\(|a| > 1\)) was used to remove points where the match surface is nearly flat, i.e., the point position is unstable.

### 3.3.2 Warping-Based Image Fusion for Polarization Computations

We turn now to the use of the imaging-consistent algorithms and the integrating resamplers for our second application, warping-based image fusion for polarization computations. These polarization computations might be used as a preprocessing step for other algorithms, e.g., as a preprocessing step for removal of specularities using color and polarization.

In [Wolff and Boult-1991, Nayar et al.-1993], the experimental setup used a linear polarizing filter mounted on a precision rotation ring and placed in front of the camera lens. Images are taken at various orientations of the polarizer. This setup requires manually adjusting the orientation of the linear polarizing filter, which is impossible for applications involving motion. More recent work by Wolff [Wolff-1994] allows a LCD shutter to automate the polarization orientations, but it still requires 1/5 of a second to acquire the polarization information, limiting its use to static scenes. The accuracy of the computation of the percent polarization images depends, to a large extent, on how accurately the images are aligned and the accuracy of the intensity values after warping. The goal of our warping-based technique is to support polarization computations on moving objects.
Rather than using a single camera and a single polarizing filter, we have four Sony XC-999/999P color cameras tightly bundled together. Each has a linear polarizing filter mounted in front of its lens. Thus in a single frame time, we can acquire 4 images at different polarization orientations. However, this setup introduces new problems. How do we align the images taken from these cameras? Warping! And, how do we ensure that the intensity values are accurate after the images are aligned? The integrating resampler, of course. (Note that with a very-short baseline (approximately 2.5cm) relative to the work area (approximately 25cm), and a moderate scene distance (approximately 2m), the variations due to scene depth are expected to be quite small. We calibrate the warping between the images using a fixed pattern. Objects are simply placed into the scene.

Figure 3.5 shows the test images before they are warped to the reference image; Figure 3.6 the test images after they are warped to the reference image. More details can be found in [Zhou-1995].

Again, we use the integrating resamplers described herein to solve this problem. For this experiment, we choose one of the four images as a reference image and warp the others to that reference image. Camera calibration, especially radiometric calibration, is needed for polarization computations (even with a single camera).

To evaluate its use in polarization, we set up the cameras using a rotating polarizer as in our earlier work. Using this, we could compute polarization using a single camera and via warping and thus compute the difference, which we have considered “errors”. We note that “random” variations on the order of 5-10% of the computed polarization values occur when using a single camera setting. While there are numerous ways we could analyze this, we broke the error computations into 5 regions, shown in Table 3.2 (See also [Zhou-1995, Table 3.12]). The regions consider high gradient of polarization
Figure 3.5: Images for polarization example. (a) reference image; (b)–(d) images before they are warped to the reference image.

as well as high and low polarization. In that table, $P$ refers to the percent polarization at a pixel, and $\nabla P$ means the gradient of the percent polarization image at that point.
Figure 3.6: Images for polarization example. (a) reference image; (b)–(d) images after they are warped to the reference image.

As can be seen in Table 3.2, the errors in warping-based polarization are quite acceptable. The larger errors in regions of very low polarization and regions of high
polarization gradient are consistent with variations in polarization from a single camera, where those two regions have larger errors. We have done a few such experiments, with varying degrees of success; for example, in a close-up experiment on curved objects, some of the the polarization errors were closer to 25%. While the integrating resampler can handle the warping with sufficient accuracy, warping is just a part of the polarization computation problem. Further work is needed before the multi-camera polarization computations are robust enough for general use. Difficulties arise because of:

- camera calibration (we need accurate radiometric calibration between the 4 cameras),
- lighting variations,
- objects with high curvature (which cause specularities to move significantly for small changes in viewpoint), and
- objects with regular high frequency textures that can increase aliasing.

<table>
<thead>
<tr>
<th>Region</th>
<th>RMS</th>
<th>Avg PP</th>
<th>%Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \nabla P &lt; 10 )</td>
<td>1.12</td>
<td></td>
<td>2.7%</td>
</tr>
<tr>
<td>( 10 &lt; \nabla P &lt; 35 )</td>
<td>0.44</td>
<td></td>
<td>2.7%</td>
</tr>
<tr>
<td>( \nabla P &lt; 10, P &lt; 10 )</td>
<td>0.36</td>
<td></td>
<td>12%</td>
</tr>
<tr>
<td>( 10 &lt; \nabla P &lt; 20 )</td>
<td>0.96</td>
<td></td>
<td>5.6%</td>
</tr>
<tr>
<td>( 20 \leq \nabla P )</td>
<td>1.62</td>
<td></td>
<td>7.7%</td>
</tr>
</tbody>
</table>

Table 3.2: Errors in warping-based polarization computations.
3.4 Conclusion

This chapter introduced integrating resamplers as an efficient method for warping with the imaging-consistent algorithms. Examples show the usefulness of integrating resamplers in two low-level computer vision applications. Evaluations were made by comparing the resulting images using the integrating resamplers and those using bi-linear interpolation. If accuracy in the underlying intensity values matters, the integrating resampler offers advantages over traditional bi-linear mapping.
Chapter 4

Image-Based Super-Resolution

This chapter introduces a new algorithm for enhancing image resolution from an image sequence. The approach we propose herein uses the integrating resampler described in Chapter 3 as the underlying resampling algorithm. Moreover, it is a direct method, which is fundamentally different from the iterative, back-projection approaches proposed in [Peleg et al.-1987, Keren et al.-1988, Irani and Peleg-1991, Irani and Peleg-1993, Bascle et al.-1996]. We show that image warping techniques may have a strong impact on the quality of image resolution enhancement. By coupling the degradation model of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real sensors, which also improves the quality of super-resolution images. Examples of super-resolutions are given for gray-scale images. Evaluations are made by comparing the resulting images and those using bi-linear resampling and back-projection in the spatial and frequency domain. Results from our experiments show that integrating resampler outperforms traditional bi-linear resampling.
4.1 Introduction

Earlier research on super-resolution dates back to the work by Huang and Tsai [Huang and Tsai-1984]. They solved the problem in the frequency domain, but they disregarded sensor blurring. Furthermore, they assumed only inter-frame translations. Gross [Gross-1986] solved the problem by first merging the low resolution images by interpolation and then obtaining the high-resolution image by deblurring the merged image. Again, only inter-frame translations were considered. Peleg and Keren [Peleg et al.-1987, Keren et al.-1988] solved the problem by first estimating an initial guess of the high-resolution image, then simulating the imaging process and minimizing the difference between the observed and simulated low-resolution images. Irani and Peleg [Irani and Peleg-1991, Irani and Peleg-1993] used a back-projection method similar to that used in tomography to minimize the same difference. Bascle et al. [Bascle et al.-1996] used the same back-projection method to minimize the difference between the predicted and actual images except that a simple motion blur model was included. All previous work ignores the impact of image warping techniques. In this chapter, we show that warping techniques may have a strong impact on the quality of image resolution enhancement.

Image warping has been around almost as long as image processing, allowing users to reshape the image geometry. Image warping requires the underlying image to be “resampled” at non-integer locations; hence, it requires image reconstruction. When the goal of warping is to produce output for human viewing, only moderately accurate image intensities are needed. In these cases, techniques using bi-linear interpolation have been found sufficient. However, as a step for applications such as super-resolution, the precision of the warped intensity values is often important. For these problems, bi-linear image reconstruction may not be sufficient.
This chapter shows how the integrating resampler can be used to enhance image resolution. A detail description of the integrating resampler can be found in Chapter 3. The image formation process as well as the sensor model are reviewed in Chapter 2. We therefore proceed directly to the super-resolution algorithm where we compare the integrating resampler with bi-linear resampling.

4.2 Image-Based Super-Resolution

We turn now to the problem of super-resolution from an image sequence. Super-resolution refers to the process of constructing high-resolution images from low-resolution image sequences. Given the image sequence, our super-resolution algorithm is now formulated, as follows:

1. Choose one of the images as the reference image, and compute the motion field between all the images and the reference image.

2. Scale up the reference image, and then warp all the images to the reference image based on the motion field computed in the previous step and the chosen scale.

3. Obtain a super-resolution image by fusing all the images together.

4. Deblur the resulting super-resolution image.

The idea of super-resolution is based on the fact that each image in the sequence provides small amount of additional information. By warping all the images to the reference image (scaling at the same time the images are being warped) and then fusing together all the information available from each image, a super-resolution image can be constructed.
It is worth mention in passing that just increasing the sampling rate is *not* super-resolution because it does not provide additional information. For instance, scaling up an image by a factor of 2 is not super-resolution no matter what interpolation techniques are used. The central idea of super-resolution is to bring together additional information available from each image in the image sequence. Even if the final sampling rate is exactly the same as the original, we can still do super-resolution. In this case, it provides a sharper image of the original size. As an example, we demonstrate this in Figure 4.1.

We presume that “motion” is computed, which is easy for rigid transformation. For this example, the motion field computation is based on a normalized correlation matching algorithm. The size of the template window is 7x7 for the first three experiments. The result of the matching is a dense disparity surface. The sub-pixel accuracy of matching was achieved by fitting a quadratic (i.e., \( ax^2 + bx + c \)) through the point with a unique match and its two neighbors. This means solving a linear system with a 3x3 Vandermonde’s matrix as the coefficient matrix, the coefficients of the quadratic as the unknowns, and the disparities as the right-hand side. If the match is not unique or if the match is unique but has only one neighbor, then that point is not used. Also, heuristics (\(|a| > 1\)) was used to remove points where the match surface is nearly flat, i.e., the point position is unstable. The \(x\) and \(y\) maps are obtained by fitting a plane to the disparity data.

When off-the-shelf lenses and cameras are used, pre-warping can be used to remove the distortions (See [Tsai-1986, Tsai-1987, Chiang and Boult-1995] and Appendix A). In Chapter 3, we showed that the integrating resampler can improve the quality of the matching.
We observed that the number of images required depend on the scaling factor. In general, the larger the scaling factor, the more images are needed. We also tried several different approaches to fuse the images together, including averaging and median filters. Our experiments show that the median filter is better, though often not much better than the averaging filter.

The test data was taken using two different Sony cameras, XC-77 and XC-999, attached to a Datacube MV200 System. Figures 4.1, 4.2, and 4.3 show our first experimental results. Figure 4.6 shows our second experimental results. The resulting super-resolution images are 256x256, i.e., a scale-up by a factor of 4. Shown are the central 60x60 and 240x240 pixels. We note that previous work on this topic reported results only scaling by a factor of 2.

We work on a scale-up by a factor of 4 for two reasons:

1. Increasing the resolution increases the distance between the replication in the frequency domain, thus making the reconstruction process less sensitive to the imperfectness of the reconstruction kernel.

2. Increasing the resolution also reduces the sensitivity of the shape of the reconstruction kernel at the ends because the signal to the ends of the kernel is much smaller compared to a scale-up by a factor of 2.

We may also choose to work on a scale-up by a factor larger than 4, but the gain is relatively small compared to by a factor of 4. This is consistent with the results shown in Section 4.3.

1Because of the limited resolution and the limited number of shades supported by the printer, the visible quality of the images shown in this chapter may not reflect the actual difference. We recommend that you obtain electronic copy and view on your monitor.
Figure 4.1: Super-resolution results at the input sampling rate. See page 63 for details.

Figure 4.1 shows some super-resolution results at the sampling rate of the original images. Figure 4.1a shows one of the eight original images. Figure 4.1b shows
super-resolution using bi-linear resampling followed by down-sampling using QRW; Figure 4.1c, super-resolution using QRW followed by down-sampling using QRW. Figure 4.1d shows super-resolution by back-projection using bi-linear resampling followed by down-sampling using bi-linear resampling. Figure 4.1e shows super-resolution using bi-linear resampling followed by deblurring followed by down-sampling using QRW; Figure 4.1f, super-resolution using QRW followed by deblurring followed by down-sampling using QRW. Figure 4.1g shows the deblurred version of Figure 4.1a. The next two figures show the importance of deblurring after super-resolution rather than applying super-resolution to deblurred originals. Figure 4.1h shows deblurring followed by super-resolution using bi-linear resampling followed by down-sampling using QRW; Figure 4.1i, deblurring followed by super-resolution using QRW followed by down-sampling using QRW. It can be easily seen from Figure 4.1 that image warping techniques indeed have a strong impact on the quality of image resolution enhancement. By coupling the degradation model of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real sensors and highly improve the quality of super-resolution imaging. In particular, Figure 4.1f is significantly clearer than the original (Figure 4.1a) or a deblurred version thereof (Figure 4.1g). Thus, super-resolution provides added benefits even if the final sampling rate is exactly the same as the original.

Figure 4.2 shows the final results of our first experiment. Figure 4.2a shows Figure 4.1a blown up by a factor of 4 using pixel replication. Figure 4.2b shows super-resolution by back-projection using bi-linear resampling to simulate the image formation process and Figure 4.3a as the initial guess. Figure 4.2c shows super-resolution using bi-linear resampling followed by deblurring; Figure 4.2d, super-resolution using QRW
Figure 4.2: Final results from an 8x64x64 image sequence taken by XC-77. (a) Figure 4.1a blown up by a factor of 4 using pixel replication; (b) super-resolution by back-projection using bi-linear resampling to simulate the image formation process and Figure 4.3a as the initial guess; (c) super-resolution using bi-linear resampling followed by deblurring; (d) super-resolution using QRW followed by deblurring. Note that (b) shows the results based on our implementation of Irani’s back-projection method.
followed by deblurring. Note that Figure 4.2b shows the results based on our implementation of the back-projection method described in [Irani and Peleg-1991]. Also, for the purpose of comparison, we are assuming that Figures 4.2c and 4.2d have undergone the same degradation before sampling. Figure 4.3 shows the results without deblurring.

For the purpose of comparison, Figures 4.4 and 4.5 show the same results as shown in Figure 4.2 except that all the images are normalized with respect to the super-resolution result shown in Figure 4.2d (repeated in Figures 4.4d and 4.5d for the convenience of comparison). Figure 4.4 shows the results after all the images are normalized so that the means are identical, as follows:

\[
I_n = \frac{\bar{I}_s}{\bar{I}_u} I_u
\]

where \(I_n\) and \(I_u\) are, respectively, the normalized image and the image to be normalized, and \(\bar{I}_s\) and \(\bar{I}_u\) are, respectively, the average of the intensity values of the super-resolution
Figure 4.4: Same results as shown in Figure 4.2 after being normalized to have the same mean value.

image shown in Figure 4.2d and the average of the intensity values of the images to be normalized, i.e., the images shown in Figures 4.2a, b, and c.
Figure 4.5: Same results as shown in Figure 4.2 after being normalized to have the same dynamic range.

Figure 4.5 shows the results after all the images are normalized so that the dynamic ranges are identical, as follows:

\[ I_n = \min_i + \frac{\max_s - \min_i}{\max_u - \min_u} (I_u - \min_u) \]
where $I_n$ and $I_u$ are, respectively, the normalized image and the image to be normalized, $\max_s$ and $\min_s$ are, respectively, the minimum and maximum intensity values of the super-resolution image shown in Figure 4.2d, and $\max_u$ and $\min_u$ are, respectively, the minimum and maximum intensity values of the images to be normalized, i.e., the images shown in Figures 4.2a, b, and c. In this particular case, since the values of $\min_s$ and $\max_s$ are, respectively, 0 and 255, the images are eventually normalized with respective to the maximum dynamic range, i.e., the range from 0 to 255.

Figure 4.6: Final results from a 3264x64 very noisy image sequence taken by XC-999. (a) one of the original images blown up by a factor of 4; (b) super-resolution using QRW followed by deblurring.

Figure 4.6 shows the final results of our second experiment, this time using a Sony XC999, which is a one-chip color camera. The “pixel” artifacts are a result of color quantization. Figure 4.6a shows one of the original images blown up by a factor of 4; it can be easily seen that inter-frame motion is involved in this case. Figure 4.6b shows super-resolution using QRW followed by deblurring. Obviously, our super-resolution
method gets rid of most of the artifacts due to the image formation process and the motion as well.

Figure 4.7 shows the final results of our third experiment, this time using a grayscale image. Figure 4.7a shows one of the original images blown up by a factor of 4 using QRW without deblurring; Figure 4.7b, deblurred version of Figure 4.7a. Figure 4.7c shows super-resolution using QRW without deblurring; Figure 4.7d, deblurred version of Figure 4.7c. Again, our super-resolution method gets rid of most of the artifacts due to the image formation process and the motion as well.

Fig. 4.8 shows our fourth example—a more complex gray level image. Fig. 4.8a shows one of the original images pixel replicated by a factor of 4; Fig. 4.8b super-resolution using QRW followed by deblurring. The tread-wheels of the toy tank are visible in the super-resolution image but not in the originals, and the “tank-number” is (just) readable in the super-resolution image while not in the original.

We tried several normalizing factors using bi-linear resampling and chose one of the back-projected images (Figure 4.2d) that minimizes the sum-of-square difference (SSD) between the observed and simulated images. It is worth pointed out that in this particular case, SSD is not necessarily a good error measure because it is not robust. Moreover, the back-projection method proposed in [Irani and Peleg-1991] is not only computationally expensive but also difficult to normalize. Furthermore, the same normalizing factor does not always give the best result in terms of the error measure when different resampling algorithms are used.

Results from our experiments show that the direct method we propose herein is not only computationally cheaper, but it also gives results comparable to or better than those using back-projection. Moreover, it is easily seen from Figures 4.1, 4.2, and 4.3
Figure 4.7: Final results from an 8 64x64 grayscale image sequence taken by XC-77 after being normalized to have the same dynamic range. (a) one of the original images blown up by a factor of 4 using QRW without deblurring; (b) deblurred version of (a); (c) super-resolution using QRW without deblurring; (d) deblurred version of (c).
that integrating resampler outperforms traditional bi-linear resampling. Not surprisingly, our experiments show that most of the additional information carried by each image is concentrated on the high frequency part of the image. This observation also explains why the integrating resampler outperforms bi-linear resampling; as already shown in [Boult and Wolberg-1993], bi-linear resampling causes more blurring than the integrating resampler.

The running times vary from of our first experiment measured on a Sun 143MHz Ultra SPARC running Solaris 2.5 as well as a 120MHz Pentium running Linux 2.0.12 are given in Table 4.1. Note that for precision reasons, all the operations are done in double-precision floating point. We are currently working on the integer version, which we believe will tremendously speed up the whole operation. Also, note that running times required by both methods such as computation of the motion field are not included. As
shown in Table 4.1, for this example, our method is more than four times faster than our implementation of Irani’s back-projection method. In general, Irani’s back-projection method takes time roughly proportional to the number of iterations as well as the size of the degradation model. Our experiments show that though each iteration of Irani’s back-projection method takes only about two thirds the running times of our method, at least two iterations are required to determine when to stop iterating. Thus, Irani’s back-projection method is about one third slower than our method even in the best case, i.e., only two iterations are required. Our experiments also show that more than three iterations are often required to minimize the sum-of-square difference. This means that our method is often more than two or three times faster than Irani’s back-projection method.

Table 4.1 also shows the running times using bi-linear resampling. Compared to bi-linear resampling, on SPARC, QRW is about 54% slower while on Pentium, it is about 27% slower.
4.3 Spectral Analysis

In this section, we present the spectral analysis of the super-resolution algorithm described in Section 4.2. We begin with the Fourier spectra\(^2\) of our first experimental results shown in Figures 4.1, 4.2, and 4.3. We then turn our attention to super-resolution from a sequence of 5 synthetically down-sampled images where more details of spectral analysis are given.

4.3.1 Spectral Analysis of Super-Resolution from Real Images

In this subsection, we present the Fourier spectra of our first experimental results described in Section 4.2. Figure 4.11 shows the spectral results after being multiplied by a factor of \(25^3\).

Figure 4.9 shows the whole Fourier spectra of our first experimental results (See Figure 4.2). Figure 4.9a shows the Fourier spectrum of one of the original images scaled up by a factor of 4 using bi-linear resampling (no super-resolution). Figures 4.9b, c, and d show, respectively, the Fourier spectrum of the super-resolution image given in Figures 4.2b, c, and d, i.e., super-resolution using back-projection, bi-linear resampling with deblurring, and QRW with deblurring.

Figure 4.10 shows exactly the same results as shown in Figure 4.9 except that they are shown in log space\(^4\). It can be easily seen from Figure 4.10b that super-resolution using back-projection seems to cause lots of ringing near the edges.

Figures 4.11a and b show the Fourier spectra of two of the original images without

\(^2\)Note that for the purpose of microfilming, the Fourier spectra shown in this section are reversed so that darker means larger and brighter means smaller.

\(^3\)The scaling factor 25 was chosen to maximize the intensity values while at the same time minimize the number of intensity values that may go beyond the maximum intensity value 255.

\(^4\)The scaling factor 500 was chosen to make it easier to see the results.
Figure 4.9: Fourier spectra of our first experimental results shown in Section 4.2 after being multiplied by a factor of 25. (a) one of the original images scaled up using bi-linear resampling; (b) super-resolution using back-projection; (c) super-resolution using bi-linear resampling with deblurring; (d) super-resolution using QRW with deblurring.

deblurring; Figures 4.11c and d the Fourier spectra of the same two images shown in Figures 4.11a and b with deblurring. For the purpose of comparison, Figures 4.11e–h show
Figure 4.10: Display of \( \log(1 + |F(u)|) \) where \(|F(u)|\) are the Fourier spectra shown in Figure 4.9 after being multiplied up by a factor of 500. (a) one of the original images scaled up using bi-linear resampling; (b) super-resolution using back-projection; (c) super-resolution using bi-linear resampling with deblurring; (d) super-resolution using QRW with deblurring.
Figure 4.11: Fourier spectra of our first experimental results shown in Section 4.2 after being multiplied by a factor of \( \frac{1}{25} \). (a) and (b) two of the original images; (c) and (d), respectively, (a) and (b) deblurred; (e)–(h) the central 64x64 region of the Fourier spectra shown in Figure 4.9, with (e) showing the central 64x64 region of Figure 4.9a, (f) showing the central 64x64 region of Figure 4.9b, (g) showing the central 64x64 region of Figure 4.9c, and (h) showing the central 64x64 region of Figure 4.9d.

Compared to Figures 4.11a and b, Figure 4.11c and d show that deblurring increases the high frequency components that do not exist in the original image. Figure 4.11e shows that when the original high-resolution image is down-sampled and then up-sampled, lots of the high frequency components are just lost, and no resampling algorithms can recover the lost information. Figure 4.11h shows that super-resolution using QRW with deblurring outperforms super-resolution using bi-linear resampling with de-
blurring and back-projection.

4.3.2 Spectral Analysis of Super-Resolution from Synthetically Down-Sampled Images

We now turn our attention to super-resolution from a sequence of 5 synthetically down-sampled images. The original high-resolution image is real so that we can have the ground truth with which to compare the scaled-up images.

Figure 4.12 shows the experimental results. The test images are generated by first translating and then down sampling the high-resolution image shown in Figure 4.12a using QRW. Figure 4.12a shows the original high-resolution image; Figure 4.12b the scale-up of the down-sampled version of Figure 4.12a by a factor of 4 using bi-linear resampling (no super-resolution); Figure 4.12c the super-resolution result from the synthetically down-sampled image sequence using bi-linear resampling with deblurring; Figure 4.12d the super-resolution result from the synthetically down-sampled image sequence using QRW with deblurring.

Table 4.2 shows the powers of the Fourier transform of the images shown in Figure 4.12. The images are divided into regions as shown in Table 4.2a, with $P_3$ being the whole region (i.e., the whole image); $P_2$, $P_1$ and $P_0$ being the regions inside the inner squares (respectively, 192x192, 128x128, and 64x64); $P_2'$, $P_1'$, and $P_0'$ being the regions outside the inner squares. The powers of the Fourier transform of the images shown in Figure 4.12 are given in Table 4.2b. The column marked $P_3$ shows the power of the whole region (i.e., the whole image). The columns marked $P_2$, $P_1$, and $P_0$ show, respectively, the power of the central 192x192 region, the power of the central 128x128 region, and the power of the central 64x64 region. The column marked $P_2'$, $P_1'$, and $P_0'$ show, respec-
Figure 4.12: Results from a sequence of 5 synthetically down-sampled images. (a) the original image; (b) the scale-up of the down-sampled version of (a) by a factor of 4 using bi-linear resampling (no super-resolution); (c)–(d) super-resolution from the synthetically down-sampled image sequence using, respectively, bi-linear resampling with deblurring and QRW with deblurring.
tively, the power of the whole region minus the power of the central 192x192 region, the power of the whole region minus the power of the central 128x128 region, and the power of the whole region minus the power of the central 64x64 region. As can be easily seen from Table 4.2b, most of the power concentrates in the central 64x64 region. Outside this region, the power is relatively small. Obviously, super-resolution using QRW with deblurring does considerably better!

\[ P_3 \quad P_2 \quad P_1 \quad P_0 \]

\[ P'_3 \quad P'_2 \quad P'_1 \quad P'_0 \]

Table 4.2: Power of the Fourier transform of the images shown in Figure 4.12. (a) Regions involved in the computation of the powers of the Fourier transform shown in (b), with \( P_3 \) being the whole region (i.e., the whole image); \( P_2, P_1 \) and \( P_0 \) being the regions inside the inner squares (respectively, 192x192, 128x128, and 64x64); \( P'_2, P'_1 \), and \( P'_0 \) being the regions outside the inner squares. (b) Power of the Fourier transform of the images shown in Figure 4.12 corresponding to the regions shown in (a), i.e., \( P_3, P_2, P_1, P_0, P'_2, P'_1 \), and \( P'_0 \) showing, respectively, the power of the whole region. the power of the central 192x192 region, the power of the central 128x128 region, the power of the central 64x64 region, the power of the whole region minus the power of the central 192x192 region, the power of the whole region minus the power of the central 128x128 region, and the power of the whole region minus the power of the central 64x64 region.

Figures 4.13, 4.14, and 4.15 show the Fourier spectra of scanlines 128, 100, and
Figure 4.13: Display of $\log(1 + |F_{128}(u)|)$ where $|F_{128}(u)|$ are the Fourier spectra at the DC component in $y$ of the images shown in Figure 4.12.

of the corresponding images shown in Figure 4.12 in log space, i.e., $\log(1 + F_n(u))$ where $F_n(u)$ is the Fourier spectrum of scanline $n$. Figure 4.13 shows scanline 128, the
Figure 4.14: Display of $\log(1 + |F_{100}(u)|)$ where $|F_{100}(u)|$ are the Fourier spectra near the Nyquist rate of the images shown in Figure 4.12.

DC component in $y$ showing all $x$ variations; Figure 4.14 scanline 100, near the Nyquist rate in $y$ showing the $x$ variations; Figure 4.15 scanline 92, just beyond the Nyquist rate
Figure 4.15: Display of $\log(1 + |F_\omega(u)|)$ where $|F_\omega(u)|$ are the Fourier spectra just above the Nyquist rate of the images shown in Figure 4.12.

in $y$ showing the $x$ variations. It can be easily seen from Figures 4.13, 4.14, and 4.15 that super-resolution using QRW with deblurring gives result that is closest to the original
high-resolution image, i.e., the ground truth. They also show that outside the region [64, 192], the Fourier spectrum is close to zero. This is consistent with the powers shown in Table 4.2.

Figure 4.16: Display of $|F_{128}^S(u)| - |F_{128}^O(u)|$ where $|F_{128}^S(u)|$ and $|F_{128}^O(u)|$ are the Fourier spectra at the DC component in $y$ of the images shown in Figure 4.12.

Figures 4.16, 4.17, and 4.18 show the difference between the Fourier spectra of scanlines 128, 100, and 92 of the images shown in Figures 4.12b, c, and d and the Fourier spectra of scanlines 128, 100, and 92 of the ground truth high-resolution image shown in Figure 4.12a, i.e., $|F_n^S(u)| - |F_n^O(u)|$ where $F_n^S(u)$ and $F_n^O(u)$ are, respectively,
Figure 4.17: Display of $|F_{100}^S(u)| - |F_{100}^O(u)|$ where $|F_{100}^S(u)|$ and $|F_{100}^O(u)|$ are the Fourier spectra near the Nyquist rate of the images shown in Figure 4.12.

the Fourier spectrum of scanline $n$ of the images shown in Figures 4.12b, c, and d and the Fourier spectrum of the original high-resolution image shown in Figure 4.12a. Figure 4.16 shows scanline 128; Figure 4.17 scanline 100; Figure 4.18 scanline 92, with a showing the difference between the scale-up of the down-sampled version of the original by a factor of 4 using bi-linear resampling (no super-resolution) and the original; b showing the difference between super-resolution using bi-linear resampling with de-blurring and the original; c showing the difference between super-resolution using QRW
Figure 4.18: Display of $|F_{92}(u)| - |F_{92}^o(u)|$ where $|F_{92}(u)|$ and $|F_{92}^o(u)|$ are the Fourier spectra just above the Nyquist rate of the images shown in Figure 4.12.

with deblurring and the original. The results show that super-resolution using QRW with deblurring outperforms super-resolution using bi-linear resampling with deblurring.

4.4 Conclusion

This chapter introduced a new algorithm for enhancing image resolution from an image sequence. We show that image warping techniques may have a strong impact on the
quality of image resolution enhancement. By coupling the degradation model of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real lenses, and this, in turn, significantly improves the quality of super-resolution images. Examples of super-resolutions for gray-scale images show the usefulness of the integrating resampler in applications like this, scaling by a factor of 4 using 5–32 images. Evaluations were made by comparing, in both the spatial and frequency domains, the new techniques with results from bi-linear resampling and back-projection. Results from our experiments showed that integrating resampler outperforms traditional bi-linear resampling. We also demonstrated that super-resolution provides added benefits even if the final sampling rate is exactly the same as the original.
Chapter 5

Edge-Based Super-Resolution

Until now, all super-resolution algorithms have presumed that the images were taken under the same illumination conditions. This chapter introduces a new approach to super-resolution—based on edge models and a local blur estimate—which circumvents these difficulties. The chapter presents the theory and the experimental results using the new approach.

5.1 Introduction

We have recently proposed a new algorithm for enhancing image resolution from an image sequence (Chapter 4) where we compared our super-resolution results with those using bi-linear resampling and Irani’s back-projection method. We showed that the integrating resampler (Chapter 3) can be used to enhance image resolution. We further showed that warping techniques can have a strong impact on the quality of the super-resolution imaging. However, left unaddressed in Chapter 4 are several important issues. Of particular interest is lighting variation. The objective of this chapter is to address
techniques to deal with these issues. The examples herein use text because the high frequency information highlights the difference in super-resolution algorithms and because it allows us to ignore, for the time being, the issues of 3D edges under different views. Clearly, matching is a critical component if this is to be used for fusing 3D objects under different viewing/lighting conditions. This chapter, however, concentrates on how well we can combine them presuming good matching information.

In what follows in this chapter, we first present the new super-resolution algorithm where we compare the resulting super-resolution images with those presented in Chapter 4 in Section 5.2. The new algorithms for edge localization and local blur estimation are given, respectively, in Sections 5.3 and 5.4.

5.2 Edge-Based Super-Resolution

For many applications involving an image sequence, the problem of lighting variation arises. It is very unlikely that lighting conditions remain the same for the entire image sequence, even when they are taken consecutively in a well controlled environment. If the images are not from a short time span, we know that variations are often significant. There exist all kinds of possibilities that may cause lighting to vary, e.g., viewpoint variation, temperature of light bulbs, clouds or people passing by, and so on. Thus, the problem is that how do we eliminate this undesirable effect? One of the easiest solutions to this problem is, of course, just to ignore lighting variations and assume that the effect of lighting variations is negligible. However, this is limiting. The idea we propose herein is a simple solution. Rather than obtaining a super-resolution image by fusing all the images together, we may choose one, and only one, of the images from the image
sequence to determine the lighting and then fuse together all the edge models from the other images. This does not necessarily solve the lighting variation problem. But, this effectively mitigates the problem of lighting variation since we are now dealing with a single image as well as the edge positions that are less sensitive to the change of lighting.

However, to fuse all the edges together, it requires that the edges be first detected and then warped. If we are going to fuse edge information, we need to know the shape of the edge, which depends on the blurring. Thus, it also requires that the reference image be reestimated and scaled up based on the edge models and local blur estimation. Therefore, we generalize the idea of the imaging-consistent reconstruction/restoration algorithms to deal with discontinuities in an image.

This section is an overview and example. The new algorithm for edge detection is given in Section 5.3. The new algorithm for local blur estimation for each edge point is presented in Section 5.4. The idea of edge-based super-resolution described herein is to fuse all the edge models together.

We turn now to the problem of super-resolution from an image sequence. The idea of super-resolution is based on the fact that each image in the sequence provides small amount of additional information. By warping all the images to the reference image (scaling at the same time the images are being warped) and then fusing together all the information available from each image, a super-resolution image can be constructed. Given the image sequence, our super-resolution algorithm is now formulated, as follows:

1. Estimate the motions involved in the image sequence.

2. Estimate the edges using the procedure derived in Section 5.3.

3. Estimate the blur models for each edge point in the image using the procedure
derived in Section 5.4.

4. Choose a “reference” image to determine the lighting for the output image.

5. Warp all the edge/blur models to the reference image and fuse them.

6. Use the fused edge/blur models and the reference image to compute the super-resolution intensity image.

7. Optional deblurring stage.

In what follows, we start with examples and then explain the details.

We presume that “motion” is computed, which is easy for rigid transformation, although interlacing issues must be properly addressed.

When off-the-shelf lenses and cameras are used, pre-warping can be used to remove the distortions (See [Tsai-1986, Tsai-1987, Chiang and Boult-1995] and Appendix A). In Chapter 4, we showed that the integrating resampler can improve the quality of the matching.

We tried several different approaches to fuse the edge/blur models together, including the averaging and the median filters. Our experiments show that the median filter is better, though often not much better than the averaging filter.

The test data was taken using Sony XC-77, attached to a Datacube MV200 System. Figures 5.1 and 5.2 show our experimental results. A 32 image sequence of size 77x43 was used to compute the super-resolution image of size 308x172, i.e., a scale-up by a factor of 4.

Figure 5.1 shows two of the original images with two different illumination conditions blown up by a factor of 4 using, respectively, pixel replication and bi-linear re-
Figure 5.1: Two of the original images showing two different illumination conditions blown up by a factor of 4. (a) and (b) using pixel replication; (c) and (d) using bi-linear resampling.

Figures 5.1a and b shows the results using pixel replication; Figures 5.1c and d, the results using bi-linear resampling.

Figure 5.2 shows the final results of our experiment. Figure 5.2a shows the result-
Figure 5.2: Final results from a 32 77x43 image sequence taken by XC-77. (a) super-resolution using the algorithm proposed in Chapter 4 without deblurring at the end; (b) (a) with deblurring at the end; (c) super-resolution using the algorithm proposed herein without deblurring at the end; (d) (c) with deblurring at the end.
the resulting super-resolution image using the algorithm proposed herein without deblurring at the end; Figure 5.2d shows Figure 5.2c with deblurring at the end.

Figure 5.3 shows the results of a more complex example. Figures 5.3a and b are two of the original images showing two different illumination conditions blown up by a factor of 4 using pixel replication; Figures 5.3c and d super-resolution using, respectively, Figure 5.3a and Figure 5.3b as the reference image and the algorithm proposed herein. This example also shows that even if the object is not moving, shadow may cause a serious problem when lighting varies. In this particular case, the shadow problem makes it very difficult to fuse 3D edge models.

However, as can be easily seen from Figure 5.2, the new super-resolution algorithm proposed herein circumvents the difficulties caused by the illumination conditions while all previous algorithms simply ignore the changes in illumination. Furthermore, the running time of our method is often more than two or three times faster than Irani’s back-projection method. See Chapter 4 for the comparison between our algorithm and Irani’s back-projection method.

5.3 Edge Localization

Typically, edge detection involves the estimation of first or second derivatives of the luminance function, followed by selection of extrema or zero crossing. The edge detection algorithm we propose herein uses pretty much the same idea except that it is based on the functional form for the imaging-consistent reconstruction/restoration algorithms. Thus, its sub-pixel accuracy and other parameters are easily combined with the imaging-consistent reconstruction/restoration algorithms to provide for a reconstruction algorithm
Figure 5.3: 3D edge-based example. (a) and (b) two of the original images showing two different illumination conditions blown up by a factor of 4 using pixel replication; (c) and (d) super-resolution using, respectively, (a) and (b) as the reference image and the algorithm proposed herein.
that supports intensity discontinuities.

In what follows in this section, we present only one dimensional case; higher dimensions are treated separably. Also, we consider only the imaging-consistent reconstruction algorithm QRR derived in [Boult and Wolberg-1993]. The imaging-consistent reconstruction algorithm QRR is a cubic polynomial that spans from the center of one input pixel to the next and is given by

\[
Q_i(x) = (E_{i+2} - E_i - 2(V_{i+1} - V_i))x^3 + (2E_i - E_{i+1} - E_{i+2} + 3(V_{i+1} - V_i))x^2 + (E_{i+1} - E_i)x + V_i
\]

where \(0 \leq x \leq 1\).

To simplify our discussions, throughout this section, we use

\[
Q(x) = ax^3 + bx^2 + cx + d
\]

to represent the imaging-consistent reconstruction algorithm QRR. For convenience, we also drop the subscripts for \(Q\), \(E\), and \(V\).

Given \(Q(x)\), our edge detection algorithm is outlined below.

1. **Find the function that gives all the slopes of \(Q(x)\).** Since we have a functional form for \(Q(x)\), this is nothing more than taking the first derivative of \(Q(x)\). Clearly,

\[
Q'(x) = 3ax^2 + 2bx + c.
\]

For convenience, let us call this quadratic the slope function \(S(x)\).

2. **Find edge position with sub-pixel accuracy.** This is exactly the same as finding either zero crossing or the extremum of the slope function \(S(x)\). Again, this is
easy to derive. Clearly, the first and second derivatives of the slope function $S(x)$ are

$$S'(x) = 6ax + 2b \quad \text{and} \quad S''(x) = 6a.$$  

If $a \neq 0$, the extremum (i.e., the step edge) can be found located at $x_0 = -b/3a$, and its value is given by $S(x_0) = -b^2/3a + c$. If $x_0$ is turned out to be located outside the range $(0,1)$, it is simply assumed to be undefined. If $a = 0$, then there are two cases:

(a) If $b \neq 0$, then the extremum must occur at one end or the other of that pixel, i.e., $x_0 = 0$ or $x_0 = 1$, Its value is given by either $S(0) = c$ or $S(1) = 3a + 2b + c$.

(b) If $b = 0$, then the extremum must occur everywhere within that pixel. This is exactly the same as saying that there is no extremum.

These two cases are unimportant and therefore not considered any further.

3. **Find edge direction and magnitude.** This is easy to determine given that we know the edges in the $x$ and $y$ directions. Let $s_x$ and $s_y$ denote, respectively, the slopes of an edge in the $x$ and $y$ directions. Then, the direction and magnitude of an edge are given, respectively, by

$$E_d = \begin{bmatrix} s_x \\ s_y \end{bmatrix} \quad \text{and} \quad E_m = \sqrt{s_x^2 + s_y^2}.$$  

In the case that $x_0$ is located outside the range $(0,1)$ or $a = 0$, it can simply be assumed to be undefined.

One of the problems with edge detection is that how do we distinguish significant edges from insignificant ones? Our experiments use a predefined threshold to get rid
of insignificant edges. We recognized that thresholding is a pretty limited approach. But since the edges come from a sequence of images, some of them may be detected and some of them may be undetected. The median filtering will still find most of the edges; thus, the impact may not be that strong. Our experiments so far have found some sensitivity to the threshold but have no difficulty setting it.

### 5.4 Local Blur Estimation

In Chapter 4, we showed that deblurring after image fusion is most effective for super-resolution imaging. However, this presumes that the blur is known. In this section, we propose an algorithm for local blur estimation. The idea of this algorithm is to model a blurred edge with two components: a step edge and a blur kernel. The blurred edge can be obtained by applying the blur kernel to the step edge.

Given $Q(x)$ and the location of the blurred edge $x_0$, our local blur estimation algorithm is outlined below.

![Edge model](image)

**Figure 5.4: Edge model. See text for details.**

1. *Determine the edge model.* Before we can discuss the local blur estimation algorithm, we must specify the edge model. Figure 5.4 shows our edge model. We
model an edge as a step function \( v + \delta u(x) \) where \( v \) is the unknown intensity value and \( \delta \) is the unknown amplitude of the edge. The focal blur of this edge is modeled by a “truncated” Gaussian blur kernel.

\[
G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left( -\frac{x^2}{2\sigma^2} \right)
\]

where \( \sigma \) is the unknown standard deviation. The square wave is already accounted for in the image reconstruction techniques and is not affected by lighting. Lens blur, however, can vary with aperture and lighting. Thus, we use a smoothly varying Gaussian function as our blurring function. The complete edge model is thus given by

\[
B(x) = \begin{cases} 
    \int_{x-\alpha}^{x+\alpha} vG(x-z) \, dz, & x < x_0 - \alpha \\
    \int_{x-\alpha}^{x_0} vG(x-z) \, dz + \int_{x_0}^{x+\alpha} (v + \delta)G(x-z) \, dz, & x_0 - \alpha \leq x \leq x_0 + \alpha \\
    \int_{x-\alpha}^{x+\alpha} (v + \delta)G(x-z) \, dz, & x > x_0 + \alpha
\end{cases}
\]

Expanding and simplifying gives

\[
B(x) = \begin{cases} 
    v \text{erf} \left( \frac{\alpha}{\sqrt{2\sigma}} \right), & x < x_0 - \alpha \\
    (v + \delta) \frac{\alpha}{\sqrt{2\sigma}} + \frac{\delta}{2} \text{erf} \left( \frac{x - x_0}{\sqrt{2\sigma}} \right), & x_0 - \alpha \leq x \leq x_0 + \alpha \\
    (v + \delta) \frac{\alpha}{\sqrt{2\sigma}}, & x > x_0 + \alpha
\end{cases}
\]

where \( \alpha \) is a predefined nonnegative constant, and \( x_0 \in [0, 1] \) is the location of the step edge. Since there are three unknowns, \( v, \delta, \) and \( \sigma \), we need three constraints to solve it. In what follows, we consider only the edge model defined on the interval \([x_0 - \alpha, x_0 + \alpha]\) because outside the range, they are constant.

2. **Find solutions for the three unknowns** \( v, \delta, \) and \( \sigma \). Since we have exactly three
the system of equations can be easily solved. Note that we use the third derivative instead of the second derivative for solving the system of equations. This is to ensure that the general edge model gives exactly the same edge model located at \( x_0 \). Solving the system of equations for the unknowns \( v, \delta, \) and \( \sigma \), and noting that the value of \( \sigma \) must be positive, we have

\[
\sigma = +\sqrt{\frac{r + \sqrt{s}}{2}}
\]

\[
\delta = \left( \sqrt{2\pi} \sigma Q'(x) \right) / \exp \left( \frac{-(x - x_0)^2}{2\sigma^2} \right)
\]

\[
v = \left( Q(x) - \frac{\delta}{2} \text{erf} \left( \frac{x - x_0}{\sqrt{2\sigma}} \right) \right) / \text{erf} \left( \frac{\alpha}{\sqrt{2\sigma}} \right) - \frac{\delta}{2}
\]

where \( r = -Q'(x)/Q'''(x) \) and \( s = r^2 - 4r(x - x_0)^2 \). Since the value of \( \sigma \) must be positive, it follows that \( \sigma \) has a solution if and only if \( s > 0 \) and either \( r + \sqrt{s} > 0 \) or \( r - \sqrt{s} > 0 \); \( \sigma \) has two solutions if \( s > 0 \) and both \( r + \sqrt{s} > 0 \) and \( r - \sqrt{s} > 0 \). However, to ensure that the general edge model derived herein gives exactly the same edge model located at \( x_0 \), only \( r + \sqrt{s} > 0 \) yields a valid solution. Therefore, the solution of \( \sigma \) is unique and is given by

\[
\sigma = \sqrt{\frac{r + \sqrt{r^2 - 4r(x - x_0)^2}}{2}}
\]

The value of \( \delta \) may be positive or negative depending on the value of \( Q'(x) \). Letting \( x = x_0 \), we have the edge model located at \( x_0 \), as follows:

\[
\sigma = +\sqrt{-Q'(x_0)/Q'''(x_0)}.
\]
\[
\delta = \sqrt{2\pi}\sigma Q'(x_0),
\]
\[
v = Q(x_0) / \text{erf} \left( \frac{\alpha}{\sqrt{2\sigma}} \right) - \frac{\delta}{2}.
\]

The idea of super-resolution is to use the edge models derived above to reestimate the underlying images. Given the edge location \(x_0\) as well as its corresponding edge model \((v, \delta, \sigma)\), the underlying image can be reestimated as follows:

1. Replace the intensity values to the left of \(x_0\) by \(v\) and the intensity values to the right of \(x_0\) by \(v + \delta\). The replacement is no more than three pixels to the left and right so that the algorithm remains local.

2. Recompute \(E_i\) and \(E_{i+1}\) based on the intensity values given in the previous step. Let us call them \(\tilde{E}_i\) and \(\tilde{E}_{i+1}\).

3. Recompute \(\tilde{Q}_i(x)\) based on \(\tilde{E}_i\) and \(\tilde{E}_{i+1}\) while at the same time the image is being scaled up.

### 5.5 Image-Based vs. Edge-Based Super-Resolution

In this section, we briefly compare the two super-resolution algorithms proposed in Chapter 4 and Chapter 5. Both algorithms take time roughly proportional to the number of images in the image sequence, with the image-based fusion being the faster of the two, producing a 500x500 super-resolution image in a few seconds on a Ultra-sparc.

If the variation of lighting is small, such as in a controlled indoor environment, the image-based approach is more appropriate because it uses the intensity information provided by the whole image sequence to construct the super-resolution image and thus is
better at removing the noise and undesirable artifacts. On the other hand, the edge-based algorithm is more appropriate if the variation of illumination is large.

If the variation of lighting is intermediate, a possible solution is probably a hybrid of the two algorithms we propose herein. The idea is that instead of choosing a single reference image of the edge-based super-resolution algorithm, use the averaging or median of a sub-sequence out of the image sequence as the reference image, presuming that the variation of lighting is not so significant within the sub-sequence.

5.6 Conclusion

This chapter introduced a new algorithm for enhancing image resolution from an image sequence. The new algorithm is based on edge models and local blur estimation. Again, the approach we propose herein uses the integrating resampler as the underlying resampling algorithm. We address techniques to deal with lighting variation, edge localization, and local blur estimation. For the purpose of comparison, qualitative evaluations are made by comparing the resulting images with those in our previous work.
Chapter 6

Quantitative Measurement of
Super-Resolution Using OCR

To date, work in super-resolution imaging has not addressed the issue of quantitatively measuring its advantages. This chapter presents a new approach for quantitatively measuring super-resolution algorithms. The approach, which we refer to as OCR-based measurement, uses OCR as the fundamental measure. We show that even when the images are qualitatively similar, quantitative differences appear in machine processing. A more general approach, which we refer to as appearance matching and pose estimation based approach, can be found in Chapter 7.

6.1 Introduction

Chiang and Boult-1996a, Chiang and Boult-1997b]. However, none of the previous work has addressed the issue of measuring the super-resolution results quantitatively instead of visually. The objective of this chapter is to propose a new approach for quantitative measurement of super-resolution imaging.

In this chapter, we use a character recognition rate based on a commercial OCR program. We chose this because it is easy to make it quantitative, and we consider this to be a good measure for other applications such as license-plate reading and other pattern recognition problems.

As an example, this chapter shows the experimental results of quantitatively measuring the image-based super-resolution algorithm described in Chapter 4. The proposed approach is, however, not the image-based super-resolution algorithm specific. The only requirement imposed is that only text can be measured by the approach described herein.

In what follows, we first present the new approach in Section 6.2. Experimental results are described in Section 6.3. Conclusion is given in Section 6.4.

6.2 OCR Based Measurement

The fundamental idea of the approach proposed herein is, as the name suggests, to use OCR as our fundamental measure. The algorithm contains three basic steps:

1. Obtain the super-resolution images using the super-resolution algorithm described in Chapter 4.

2. Pass the super-resolution results obtained in the previous step through a “character-oriented” OCR system.

3. Determine the number of characters recognized, i.e., the rate of recognition.
Evaluations herein are made by comparing the super-resolution results and those using bi-linear resampling.

While most “state-of-the-art” OCR programs use dictionary lookup to aid in their recognition, we consider it important to use a pure character based system because its behavior is driven by the input and not significantly impacted by the grammatical context of examples. We have also chosen to use a font-independent system, i.e., one that is not trained on the font and resolution being used. Training the OCR system would allow that training to makeup for poor, but consistent, behavior in the preprocessing.

The OCR program used for the experiments described herein is “Direct for Logitech, Version 1.3.” The images used are normalized with respect to the super-resolution image with deblurring, as follows:

\[ I_n = \frac{I_s}{I_u} \]

where \( I_n \) is the normalized image, \( I_u \) is the image to be normalized, \( I_s \) is the average of the intensity values of the super-resolution image with deblurring, and \( I_u \) is the average of the intensity values of the image to be normalized. Another possibility is to use the difference (to avoid rounding errors) or the ratio of medians to normalize the images, as follows:

\[ I_n = I_u + (m_s - m_u) \quad \text{or} \quad I_n = \frac{m_s}{m_u}I_u \]

where \( m_s \) is the median of the intensity values of the super-resolution image with deblurring and \( m_u \) is the median of the intensity values of the image to be normalized. However, that would not make much difference. Except for the first example, the same threshold is used to binarize all the images.

Since the OCR program is not trained on the particular font, we break our analysis of errors up into two categories. The first error measure compares the OCR output
Table 6.1: Characters divided into categories, with category “−” indicating all characters that are visually distinguishable.

to the ground-truth input characters. We use $C^i_r$ to indicate the number of characters recognized; $C^i_u$ to indicate the number of characters unrecognized; and $%C^i_r$ to indicate the percentage of characters recognized. For many fonts, some characters are so visually similar that without the use of training or context, distinguishing pairs of characters is quite difficult, e.g., 0 vs O, 1 vs l vs ! vs |, and in some fonts, / vs l vs t and h vs b (See Figure 6.5). Thus, we compute a second “error” measure based on the visually distinguishable characters—with $C^d_r$ denoting the number of visually distinguishable characters correctly recognized, $C^d_u$ the number of visually distinguishable characters incorrectly recognized, $C^a_r$ the number of “ambiguous” characters “correctly” recognized (i.e., any one of the valid choices as shown in Table 6.1), $C^a_u$ the number of “ambiguous” characters incorrectly recognized, and $%C^{d+a}_r$ the percentage of total characters “correctly” recognized.

Table 6.1 gives the categories of characters and states where the visually ambiguous characters are occurred in the experiments. Figure 6.1 contains 3 ambiguous characters; Figure 6.3 4 ambiguous characters; Figure 6.5 17 ambiguous characters.

For convenience, we also use the abbreviations shown in Table 6.2 in what follows in this section.
### Abbreviation Table

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRX</td>
<td>bi-linear resampling without distortion correction and deblurring</td>
</tr>
<tr>
<td>BRDCD</td>
<td>bi-linear resampling with distortion correction and deblurring</td>
</tr>
<tr>
<td>SRDCD</td>
<td>super-resolution using QRW with distortion correction and deblurring</td>
</tr>
<tr>
<td>BR</td>
<td>bi-linear resampling without deblurring</td>
</tr>
<tr>
<td>BRD</td>
<td>bi-linear resampling with deblurring</td>
</tr>
<tr>
<td>SR</td>
<td>super-resolution using QRW without deblurring</td>
</tr>
<tr>
<td>SRD</td>
<td>super-resolution using QRW with deblurring</td>
</tr>
</tbody>
</table>

Table 6.2: Abbreviations for convenience.

### 6.3 Experimental Results

The test data shown in this section was taken using laboratory quality imaging systems, a Sony XC-77 camera, attached to either a Datacube MV200 System or a Matrox Meteor Capture Card. As we will see, better imaging reduces the need for super-resolution; lower quality cameras would either produce results too poor for OCR or would increase the significance of super-resolution imaging.

All examples herein are scale-up by a factor of 4, with the inputs being images so as to yield an image with character sizes within the range accepted by the OCR system. We qualitatively evaluated the approach on a wider range of fonts and imaging conditions and note the following: fonts with thinned letters, such as the “v” in Figure 6.3, tend to be broken into multiple letters. Characters in Slanted-Serif fonts tend to touch and fail to be recognized. Inter-word spacing is not handled well (and multiple spaces are ignored in our measures). The ease of finding a good threshold depends on the uniformity of the lighting, image contrast, and lens quality. The quantitative examples show a few of these features, but in general we choose examples that are not dominated by these artifacts.

Figure 6.1 shows the super-resolution results from a sequence of 32 391x19 im-
Figure 6.1: Super-resolution results from a sequence of 32 391x19 images taken by a Sony XC-77 Camera. (a) one of the original images scaled up using bi-linear resampling and without distortion correction; (b) and (c) the results after distortion correction, with (b) showing bi-linear resampling with deblurring and (c) showing super-resolution using QRW with deblurring.

<table>
<thead>
<tr>
<th>Method</th>
<th>Output of OCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRX</td>
<td>1234561990. te gu&lt;clp bow dog jumped over te sazs</td>
</tr>
<tr>
<td>BRDCD</td>
<td>:23*;56?990. X:e quick brown dog jt:Wed over the Lazi for</td>
</tr>
<tr>
<td>SRDCD</td>
<td>2234567890. The quick brown dog jumped over the lazes 'ox.</td>
</tr>
</tbody>
</table>

Figure 6.2: Output of OCR for the text shown in Figure 6.1.

<table>
<thead>
<tr>
<th>Method</th>
<th>$C_r$</th>
<th>$C_u$</th>
<th>$%C_r$</th>
<th>$C_i$</th>
<th>$C_m$</th>
<th>$C_e$</th>
<th>$C_s$</th>
<th>$C_{i+m+e+s}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRX</td>
<td>32</td>
<td>16</td>
<td>67</td>
<td>7</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>BRDCD</td>
<td>35</td>
<td>13</td>
<td>73</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>SRDCD</td>
<td>45</td>
<td>3</td>
<td>94</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

ages taken by a Sony XC-77 camera attached to a Datacube MV200 System before they are passed through OCR for recognition. Figure 6.1a shows one of the original images scaled up using bi-linear resampling and without distortion correction. Figure 6.1b and
Figure 6.3: Super-resolution results from a sequence of 8 430x75 images taken by XC-77. (a) one of the original images scaled up using bi-linear resampling; (b) (a) deblurred; (c) super-resolution using QRW with deblurring.

Figure 6.1c show the results after distortion correction, with Figures 6.1b showing bi-linear resampling with deblurring and Figures 6.1c showing super-resolution using QRW with deblurring.

Figures 6.2a shows the results of passing the super-resolution results shown in Figure 6.1 through OCR for recognition. The original text (See Figure 6.1) consists of total 48 characters, including the two periods but excluding the spaces. Columns marked $C_i$, $C_m$, $C_e$, and $C_s$ give, respectively, the number of incorrectly recognized characters, the number of missing characters, the number of extra characters, and the number of characters that are split into two or more characters.

Figure 6.2b summarizes the output of OCR given in Figure 6.2a. Because of the

---

1 Because of the limited resolution and the limited number of shades supported by the printer, the visible quality of the images shown in this chapter may not reflect the actual difference. We recommend that you obtain electronic copy and view on your monitor.
nonuniformity of the lighting in this example, each image had its own threshold which was chosen to maximize its recognition rate. Using bi-linear resampling without distortion correction and deblurring (BRX), 32 out of the 48 characters (67%) are recognized. Using bi-linear resampling with distortion correction and deblurring (BRDCD), 35 out of the 48 characters (73%) are recognized. Using the super-resolution algorithm given in Chapter 4 with deblurring (SRDCD), 45 out of the 48 characters (94%) are recognized. Compared to bi-linear resampling without distortion correction, super-resolution using QRW recognizes 27% more characters. Compared to bi-linear with distortion correction and deblurring, super-resolution using QRW recognizes 21% more of characters. With text consisting of thousands of characters, this is definitely a significant improvement.

Qualitatively, one can note the errors are concentrated on the outer edges of the
Figure 6.5: Super-resolution results from a sequence of 8 490x140 images taken by XC-77. (a) one of the original images scaled up using bi-linear resampling; (b) (a) deblurred; (c) super-resolution using QRW with deblurring.

example where there was the most significant distortions and the worst lighting. This example did not contain any font-related ambiguities.

Figure 6.3 shows the super-resolution results for an example with a better lens (much less distortion and blur) and larger characters (almost twice the size). The input was a sequence of 8 430x75 images taken by a Sony XC-77 camera attached to a Matrox Meteor Capture Card before they are passed through OCR for recognition. The original
Figure 6.6 shows the experimental results of passing the super-resolution results shown in Figure 6.3 through OCR. Compared to bi-linear resampling without deblurring (BR), 7% more of the characters are recognized. Compared to bi-linear resampling with deblurring (BRD), 2% more of the characters are recognized. 2% may mean a lot depending on applications. Counting the ambiguous characters will increase the rate of
recognition by 2% for bi-linear resampling without deblurring (BR); all others remain the same. Compared to bi-linear resampling without deblurring (BR), super-resolution with deblurring (SR) reduces the number of incorrectly recognized, missing, extra, and split characters reduces from 7 to 1.

Figure 6.5 shows a more extensive example including the impact of different font styles in the same text. Again, the image quality is good and the characters are “larger.” The input is a sequence of 8 490x140 images taken by a Sony XC-77 camera attached to a Matrox Meteor Capture Card. The original text consists of total 142 characters including one colon and sixteen periods.

Figure 6.6 shows the experimental results of passing the super-resolution results shown in Figure 6.5 through OCR. Compared to bi-linear resampling without deblurring (BR), 5% more of the characters are recognized. Compared to bi-linear resampling with deblurring (BRD), 3% more of the characters are recognized. Again, 3% may mean a lot depending on applications. Counting the ambiguous characters will increase the rate of recognition by 2% for all methods except bi-linear resampling which increases only 1%. Compared to bi-linear resampling with or without deblurring (BRD or BR), super-resolution with deblurring (SR) reduces the number of incorrectly recognized, missing, extra, and split characters reduces from 18 to 12.

The qualitative aspects of our experimental results can be summarized as follows:

1. The better the quality of the original images and the larger the input characters, the smaller the impact of super-resolution on OCR.

2. If there is no motion and minimal warping, super-resolution will not help much more than simple temporal averaging with bi-linear resampling.
3. Type style has a strong impact on the rate of recognition.

In general, the difficulties of using OCR as a measurement can be summarized as follows:

1. The rate of recognition depends, to a large extent, on the quality of the OCR programs. Better OCR programs, especially those that use dictionary lookup, could reduce the impact of low level processing.

2. If binary images are required for recognition, as most of the OCR programs do, then the rate of recognition is sensitive to the threshold used to convert gray-scale or color images to binary images. Different thresholds may give different results. While localized thresholding would help increase the rate of recognition, we have not used them here.

3. OCR programs treat their inputs as “binary”, thus as an evaluation for super-resolution techniques, it may seem to down-play the importance of accurate grayscale production at middle intensity levels. On the other hand, these intermediate levels do occur on character boundaries and may, in fact, be the driving factor in the superiority of the new algorithms.

### 6.4 Conclusion

This chapter introduced a new approach for quantitatively measuring super-resolution algorithms. The approach, which we refer to as *OCR-based measurement*, uses OCR as the fundamental measure. Our experiments showed that even in the cases where the super-resolution images are qualitatively similar, quantitative differences appear in machine
processing.
Chapter 7

Quantitative Measurement of Super-Resolution Using Appearance Matching and Pose Estimation

This chapter presents a more general approach for quantitatively measuring super-resolution algorithms. The approach uses appearance matching and pose estimation as the primary metric and image-quality metric as a secondary measure. The chapter presents the theory and experimental results. For pose estimation, our experiments found that when scaling up by a factor of 4 in each dimension, super-resolution from 8 images, using appearance matching with 32 eigenvectors, produces an average pose error of 1.40 degrees with a median error of 0 degrees while the mean of cubic convolution from 32 images, using appearance matching with 32 eigenvectors, produces an average pose error of 8.12 degrees with a median error of 2.8 degrees.
7.1 Introduction

In Chapter 6, we presented a new approach for quantitatively measuring super-resolution algorithms. The approach, which we refer to as *OCR-based measurement*, uses OCR as the fundamental measure. We show that even when the images are qualitatively similar, quantitative differences appear in machine processing. The objective of this chapter is to propose a more general approach. The approach, which we refer to as *appearance matching and pose estimation based approach*, uses appearance matching and pose estimation as the primary metric and image-quality metric as a secondary measure.

In this chapter, we use SLAM [Nene et al.-1994] as the underlying algorithm for appearance matching and pose estimation because SLAM is a well-established appearance matching and pose estimation system which is free to researchers; our quantitative approach is, however, *not* SLAM specific. Any appearance matching and pose estimation algorithm can be used as the underlying algorithm for the approach proposed herein. It is important that the method be appearance-based as the goal of super-resolution is to provide a better estimate of the ideal image. We consider pose estimation to be an important part of the approach for two reasons. First, it is a natural problem in its own right with an intuitive quantitative measure. Second, pose estimation forces the system to distinguishing among highly similar “images,” as the image for each pose is highly correlated to views from neighboring poses.

As an example, this chapter shows the experimental results of quantitatively measuring the image-based super-resolution algorithm described in Chapter 4. The proposed approach is, however, *not* image-based super-resolution specific, either. More importantly, the proposed approach assumes no prior knowledge of the contents of the acquired images. As a result, it can be used as a measurement of all super-resolution algorithms—
In what follows, we first present the new approach in Section 7.2. Experimental setup and image acquisition is then detailed in Section 7.3. Experimental results are described in Section 7.4. Discussions are given in Section 7.5. Conclusion is given in Section 7.6.

7.2 Appearance Matching and Pose Estimation Based Measurement

In this section, we first discuss the motivation of the new approach and then give the detailed algorithm. To simplify our discussions, throughout this chapter, we will denote each high-resolution image (i.e., images in the learning set) as $X_r$ and each scaled-up image (i.e., images scaled up by super-resolution or cubic convolution) as $Y_r$ where $r$ is the orientation or pose parameter in degrees. In the experiments herein, $X_r$ refers to the 256x256 image at orientation $r$ while $Y_r$ refers to the 4X magnified version of 64x64 images at orientation $r$. Note that the proposed approach is not limited to measurement of super-resolution images at any particular size. The only requirement imposed is that the scaled-up images have the same size as those in the learning set.

The fundamental idea of the approach proposed herein is to use pose estimation as our fundamental measure. In many cases, however, the various approaches will produce identical pose estimations (because of the limited sampling of parametric eigenspace). Thus, we supplement the pose estimation measure with an image-quality metric. While image-quality metrics are fraught with problems as an overall measure of quality, we

---

1We will make datasets and quantitative evaluations available on the web so that others can compare to our algorithm.
believe that their use, in the narrow context of images that already produce identical pose estimation, is reasonable. In our problem, there are two different image-quality metrics that are natural to consider. The first is to consider the $L_2$ difference between the scaled-up image and the original high-resolution training image to which it corresponds. In our experimental tables, this is reported as a Per-Pixel-Error (PPE), i.e.,

$$PPE_r = \|X_r - Y_r\|/N$$

where $N$ is the number of pixels in each training image.

Though our experimental results show that this measure gives a reasonable qualitative indication of how similar a scaled-up image is to the corresponding high-resolution image in the learning set, it is hard to ascribe much confidence in it as a quantitative measure because the original images contain frequencies so far above the Nyquist rate of the small images that the reconstruction techniques have no hope of recovering them. The intensity variations measured in these regions are a mixture of blurring and aliasing that are not modeled. Neither super-resolution nor any resampling algorithm will help recover the lost information. We know that the differences exist and won’t go away, but we don’t know what they mean. The interpretation is further complicated because the appearance matching algorithm does not use the original image, but a projection thereof. Thus, it is unclear what exactly is the meaning of the PPE, or its impact on object recognition or pose estimation.

For this reason, we introduce a different image-quality measurement, one which normalizes for the projections used in the appearance based matching. This measure is based on the distance between the eigenspace projections of the scaled-up image and the original image. This is justified by the observation that if all eigenvectors are used, then the distance between the corresponding points in the eigenspace is exactly the $L_2$ norm.
(see [Murase and Nayar-1993]). In our case, using all eigenvectors is too expensive; so we approximate it with a subspace projection. We introduce the so-called normalized $L_2$ norm (denoted $L_2^N$), which, for the purpose of the measurement proposed herein, is defined to be distance between projections divided by the number of eigenvectors. It is formally defined in Eq. (7.3). Again, we reiterate that this measure is considered only when the pose estimation is identical. In this case, the comparison of the projected scaled-up images to projection of the true image provides additional insights into image quality. If the pose estimates differ, the interpretation of this metric is more tenuous because, as we found in experiments, it is possible for the pose estimation to get better while the $L_2^N$ gets larger.

Assuming that the parametric eigenspace [Murase and Nayar-1993] is given, the $L_2^N$ norm is computed with respect to the image from the learning set at the same orientation as the image sequence used for the construction of scaled-up images as follows:

1. The image from the learning set at orientation $r$ is projected onto the eigenspace, that is,

$$u = [e_1, e_2, \ldots, e_m]^T [X_r - c], \quad (7.1)$$

where $c$ is the average of all the images in the learning set.

2. For each reconstruction technique, the scaled-up images are then projected onto the eigenspace, as follows:

$$v = [e_1, e_2, \ldots, e_m]^T [Y_r - c], \quad (7.2)$$

where $c$ is the same as above.

3. The $L_2^N$ norm is then computed using the above projections, as follows:
\[ L_2^N = ||u - v||/m, \]  

(7.3)

where \( u \) and \( v \) are as given in Eq. (7.1) and Eq. (7.2).

### 7.3 Experimental Setup

Figure 7.1 shows the setup used for automatic acquisition of object image sets. It is roughly the same as described in [Murase and Nayar-1993] except that (1) two cameras are used—one to acquire the high-resolution images, the other to acquire the low-resolution images; (2) two tripods are used to hold, respectively, the cameras and the light source, and (3) the background region is not assigned a zero intensity value. Only one light source is used. The object is placed on top, and near the center, of a motorized turntable. The program named \texttt{xmeteor} is designed to take care of rotating the turntable, taking the images, and doing the segmentation at the same time the images are taken, without the need of human intervention.

![Figure 7.1: Setup used for automatic acquisition of object image sets. The object is placed on top, and near the center, of a motorized turntable. See text for more details.](image)
The test images are taken using two Cosmicar/Pentax lenses (50mm and 12.5mm F/1.4 1” MI C-M) mounted on two Sony XC77 CCD cameras attached to a Matrox Meteor Capture Card. To ensure that these focal lengths provide objects in the low-resolution images that are about a quarter the size of those in the high-resolution images, the two cameras are tightly bundled together, one on top of the other. This is considered as a necessity of the experiments because we want to minimize the number of times the images are warped—all the images are warped once and only once. Also, to ensure that the images have about the same intensity values so that intensity normalization is not required, roughly the same apertures are used. The distance from the cameras to the object is proportional to the size of the object, to ensure that the object will fit into the object region.

For each object of interest, 200 images are taken. Of the 200 images, 72 are the high-resolution images to be used as the learning set of SLAM. These 72 images (denoted \(X_{0^\circ}, X_{5^\circ}, \ldots, X_{355^\circ}\)) are taken, respectively, at 0, 5, 10, \ldots, 355 degrees. The remaining 128 images are the low-resolution images taken, respectively, at 0, 90, 180, and 270 degrees\(^2\). Each orientation consists of an image sequence of 32 images to be used for the construction of super-resolution images as well as for the construction of images using cubic convolution corresponding to the high-resolution image at orientation \(r\). These four image sequences are jittered (by rotating the motorized turntable about 0.6 degrees back and forth) to obtain the information needed for constructing the super-resolution images.

The four high-resolution images at 0, 90, 180, and 270 degrees (i.e., \(X_{0^\circ}, X_{90^\circ}, X_{180^\circ},\) and \(X_{270^\circ}\)) are used as the base for computing the \(L^N_2\) norms. The \(L^N_2\) norms

\(^2\)We use only four orientations to save space and time, though up to 72 orientations—or more precisely, up to the number of images in the learning set—can be taken and used in the experiments.
are computed from the projection of these four images and the projection of super-resolution images and images scaled up by cubic convolution. Though this is not always true—depending on how the discrete points are interpolated, the projections of the high-resolution images are assumed to be exact for the purpose of comparison because they are images from the learning set.

For each orientation of interest, three pairs of super-resolution images are constructed each of which consists of one super-resolution image that is not deblurred and one that is deblurred. The three pairs of super-resolution images are constructed, respectively, from 8, 16, and 32 images.

When constructing image sets, we need to ensure that all images of an object are of the right size. Each digitized image is first segmented into an object region and a background region, as follows:

1. Two background images (one for the high-resolution images and the other for the low-resolution images) are first taken before the object images are taken.

2. The background images are then subtracted from all the object images, and the bounding box is computed using the subtracted image.

3. The object region is then extracted from the original image using the centroid of the bounding box as the centroid of the object region.

Note that computing the centroid of the object region directly from the subtracted images is unstable because a few outliers may drag the centroid far away from where it is supposed to be.
7.4 Experimental Results

In this section, we discuss the experimental results. Unless stated otherwise, we will use the experimental results of object image set 1 shown in Table 7.1 as an example for the discussions throughout this section. In SLAM, the parametric eigenspace is represented as a set of discrete points in the eigenspace. Pose estimation is accomplished by finding the closest points in the set and interpolating between them. Pilot experiments showed that increasing the number of samples in the parametric eigenspace only marginally changes the pose estimation. Thus, only the results using 128 samples are given.

<table>
<thead>
<tr>
<th>Object 1</th>
<th>16/128</th>
<th>32/128</th>
<th>PPE</th>
<th>Object 1</th>
<th>16/128</th>
<th>32/128</th>
<th>PPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{X}_0$</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{CC}_\mu$</td>
<td>5.6</td>
<td>292.1</td>
<td>5.6</td>
<td>163.8</td>
<td>0.132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{CC}_\sigma$</td>
<td>0.0</td>
<td>6.7</td>
<td>0.0</td>
<td>5.0</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SR}_8$</td>
<td>0.0</td>
<td>183.8</td>
<td>0.0</td>
<td>105.0</td>
<td>0.110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SR}_{16}$</td>
<td>0.0</td>
<td>184.0</td>
<td>0.0</td>
<td>105.2</td>
<td>0.110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SR}_{32}$</td>
<td>0.0</td>
<td>184.2</td>
<td>0.0</td>
<td>105.4</td>
<td>0.110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SRD}_8$</td>
<td>0.0</td>
<td>176.2</td>
<td>0.0</td>
<td>102.2</td>
<td>0.130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SRD}_{16}$</td>
<td>0.0</td>
<td>176.5</td>
<td>0.0</td>
<td>102.4</td>
<td>0.130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SRD}_{32}$</td>
<td>0.0</td>
<td>176.9</td>
<td>0.0</td>
<td>102.7</td>
<td>0.130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{X}_{90}$</td>
<td>-0.5</td>
<td>0.0</td>
<td>-0.5</td>
<td>0.0</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{CC}_\mu$</td>
<td>44.7</td>
<td>341.5</td>
<td>47.3</td>
<td>179.3</td>
<td>0.142</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{CC}_\sigma$</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.3</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SR}_8$</td>
<td>33.5</td>
<td>227.7</td>
<td>0.0</td>
<td>121.6</td>
<td>0.120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SR}_{16}$</td>
<td>33.5</td>
<td>227.7</td>
<td>0.0</td>
<td>121.6</td>
<td>0.120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SR}_{32}$</td>
<td>33.5</td>
<td>227.6</td>
<td>0.0</td>
<td>121.5</td>
<td>0.120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SRD}_8$</td>
<td>33.5</td>
<td>232.0</td>
<td>0.0</td>
<td>124.7</td>
<td>0.140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SRD}_{16}$</td>
<td>33.5</td>
<td>231.9</td>
<td>0.0</td>
<td>124.7</td>
<td>0.140</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathbf{SRD}_{32}$</td>
<td>33.5</td>
<td>231.8</td>
<td>0.0</td>
<td>124.6</td>
<td>0.140</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1: Experimental results of object image set 1.

Table 7.1 is divided into four blocks. The first block (labeled $X_{0^\circ}$) shows the results at 0 degrees; the second block ($X_{90^\circ}$) shows the results at 90 degrees, and so forth.
The number on the upper-left corner of the title block is the object image set number. The top row of the title block also shows the number of eigenvectors computed and the number of discrete points sampled. For example, 16/128 indicates 16 eigenvectors and 128 samples.

The rows marked $X_{0^\circ}$, $X_{90^\circ}$, $X_{180^\circ}$, and $X_{270^\circ}$ show the projections of the images from the learning set at 0, 90, 180, and 270 degrees, respectively. As can be easily seen, the projection is not exact except for $X_{0^\circ}$. The match at 90 degrees is off by $-0.5$ degrees; that is, instead of matching exactly object at 90 degrees as it is supposed to be, it matches object at 89.5 degrees. The match at 180 degrees is off by 1.7 degrees while the match at 270 degrees is off by 1.1 degrees. Thus, the original high-resolution data, when projected, yields an average pose estimation error of 0.825 degrees.

The columns marked $PE$ show pose error in degrees with respect to the pose estimation shown in the first row. For example, the object $X_{0^\circ}$ matches exactly the object at 0 degrees. However, cubic convolution reconstruction yields an average pose error of 5.6 degrees. Super-resolution, deblurring or not, exactly recovers the pose. Also shown is the $L_2^N$ norm and per pixel error. In general, when there is a tie in pose estimation, super-resolution still beats cubic convolution in terms of $L_2^N$ norms.

The rows marked $CC_{\mu}$ show the average measures (e.g., average pose error in degrees) using cubic convolution. The rows marked $CC_{\sigma}$ show the standard deviations of the associated quantities.

The rows marked $SR_8$, $SR_{16}$, and $SR_{32}$ show, respectively, the pose errors using 8, 16, and 32 images for the construction of the super-resolution images. The rows marked $SRD_8$, $SRD_{16}$, and $SRD_{32}$ show exactly the same thing except that the super-resolution images are deblurred at the end. Again, they are computed with respect to the pose
estimation shown in the first row.

Figure 7.2: (a)–(d) shows low-resolution images taken, respectively, at 0°, 90°, 180°, and 270° used for the experiments shown in Table 7.1.

Figure 7.2 shows low-resolution images used in the experiments shown in Table 7.1. Figures 7.2a–d show, respectively, one of the low-resolution images taken at 0°, 90°, 180°, and 270°.

Figure 7.3 shows the scaled-up images used in the experiments shown in Table 7.1. Figure 7.3a shows $X_{180°}$, i.e., the high-resolution image at 180 degrees. Figure 7.3b shows one of the low-resolution images from the image sequence at 180 degrees scaled up by cubic convolution provided by SLAM. Figure 7.3c shows super-resolution using QRW without deblurring; Figure 7.3d super-resolution using QRW with deblurring. This example shows one of the “worst cases” for super-resolution, with a pose error of 5.6 degrees. In this case, it significantly outperforms cubic convolution which has a pose error of 29.8 degrees. The magnitude of the difference in pose error is not obvious from a visual inspection of the scaled-up images. This highlights the importance of a quantitative “task-oriented” measurement. The fact that 32 eigenvectors significantly outperform 16 eigenvectors points to the importance of maintaining “higher frequency,” albeit lower amplitude, image structure.
Figure 7.3: Scaled-up images for object image set 1. See Table 7.1 for the quantitative results. (a) $X_{180^\circ}$, i.e., the high-resolution image at 180 degrees; (b) One of the low-resolution images from the image sequence at 180 degrees scaled up by cubic convolution provided by SLAM; (c) super-resolution without deblurring; (d) super-resolution with deblurring. This example shows one of the “worst cases” for super-resolution, with a pose error of 5.6 degrees. In this case, it significantly outperforms cubic convolution which has a pose error of 29.8 degrees. The magnitude of the difference in pose error is not obvious from a visual inspection of the scaled-up images. This highlights the importance of a quantitative “task-oriented” measurement. The fact that 32 eigenvectors significantly outperform 16 eigenvectors points to the importance of maintaining “higher frequency,” albeit lower amplitude, image structure.
<table>
<thead>
<tr>
<th>Object</th>
<th>16/128</th>
<th>32/128</th>
<th>PPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_{0r} )</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>( CC_{\mu} )</td>
<td>0.8</td>
<td>168.7</td>
<td>0.8</td>
</tr>
<tr>
<td>( CC_{\sigma} )</td>
<td>1.3</td>
<td>17.1</td>
<td>1.3</td>
</tr>
<tr>
<td>( SR_8 )</td>
<td>0.0</td>
<td>134.4</td>
<td>0.0</td>
</tr>
<tr>
<td>( SR_{16} )</td>
<td>0.0</td>
<td>133.0</td>
<td>0.0</td>
</tr>
<tr>
<td>( SR_{22} )</td>
<td>0.0</td>
<td>132.8</td>
<td>0.0</td>
</tr>
<tr>
<td>( SRD_8 )</td>
<td>0.0</td>
<td>131.7</td>
<td>0.0</td>
</tr>
<tr>
<td>( SRD_{16} )</td>
<td>0.0</td>
<td>130.4</td>
<td>0.0</td>
</tr>
<tr>
<td>( SRD_{32} )</td>
<td>0.0</td>
<td>130.1</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 7.2: Experimental results of object image set 2.

<table>
<thead>
<tr>
<th>Object</th>
<th>16/128</th>
<th>32/128</th>
<th>PPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X_{90r} )</td>
<td>-0.5</td>
<td>0.0</td>
<td>-0.5</td>
</tr>
<tr>
<td>( CC_{\mu} )</td>
<td>2.0</td>
<td>243.9</td>
<td>2.0</td>
</tr>
<tr>
<td>( CC_{\sigma} )</td>
<td>1.3</td>
<td>51.3</td>
<td>1.3</td>
</tr>
<tr>
<td>( SR_8 )</td>
<td>0.0</td>
<td>158.9</td>
<td>0.0</td>
</tr>
<tr>
<td>( SR_{16} )</td>
<td>0.0</td>
<td>166.2</td>
<td>0.0</td>
</tr>
<tr>
<td>( SR_{22} )</td>
<td>2.8</td>
<td>192.7</td>
<td>2.8</td>
</tr>
<tr>
<td>( SRD_8 )</td>
<td>0.0</td>
<td>151.1</td>
<td>0.0</td>
</tr>
<tr>
<td>( SRD_{16} )</td>
<td>0.0</td>
<td>159.0</td>
<td>0.0</td>
</tr>
<tr>
<td>( SRD_{32} )</td>
<td>2.8</td>
<td>188.1</td>
<td>2.8</td>
</tr>
</tbody>
</table>

Table 7.3: Experimental results of object image set 3.
7.5 Discussions

In this section, we discuss the experimental results and other issues related to the approach proposed herein.

- Compared to cubic convolution, super-resolution generally helps enhance pose es-

Table 7.4: Experimental results of object image set 4. 16/128\textsubscript{N} shows results for raw images. 16/128\textsubscript{B} shows results for images preblurred with a 3x3 box filter. 16/128\textsuperscript{\textsubscript{-}N} shows the difference.

<table>
<thead>
<tr>
<th>Object</th>
<th>16/128\textsubscript{N}</th>
<th>16/128\textsubscript{B}</th>
<th>16/128\textsuperscript{\textsubscript{-}N}</th>
</tr>
</thead>
<tbody>
<tr>
<td>X\textsuperscript{\textsubscript{SRD}}</td>
<td>1.1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>CC\textsuperscript{\textsubscript{PE}}</td>
<td>1.4</td>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td>C\textsuperscript{\textsubscript{CC}}</td>
<td>5.6</td>
<td>5.6</td>
<td>5.6</td>
</tr>
<tr>
<td>SR\textsuperscript{\textsubscript{PE}}</td>
<td>5.6</td>
<td>5.6</td>
<td>5.6</td>
</tr>
<tr>
<td>SR\textsuperscript{\textsubscript{CC}}</td>
<td>5.6</td>
<td>5.6</td>
<td>5.6</td>
</tr>
</tbody>
</table>

Figure 7.4: Images used for the experiments shown in Table 7.4, a case in which cubic convolution outperforms super-resolution (except at 0\textdegree). (a)–(d) shows low-resolution images at 0\textdegree, 90\textdegree, 180\textdegree, and 270\textdegree, respectively. As can be seen, most of the image content is undersampled or aliased high frequency text. As mentioned, blurring these images improves the \(L_2^N\) estimation for both cubic-convolution and super-resolution. Thus, it is not surprising that using super-resolution, which strives to improve the high-frequency content, loses.
IMATION. Even if the pose estimation is identical, super-resolution improves the accuracy in terms of the $L^N_2$ norm. For example, the first block in Table 7.1 shows that when using 16 eigenvectors, the $L^N_2$ goes from 292.1 down to about 184 and 176; when using 32 vectors, it goes from 163.8 down to about 105 and 102. That is about 37% improvement. Also, the first and second blocks of the object image set 3 in Table 7.3 show that when there is a tie in pose estimation, super-resolution still beats cubic convolution in terms of $L^N_2$ norms.

- Sometimes, super-resolution without deblurring gives a better result than that with deblurring in terms of the $L^N_2$ norms. For example, the third block in Table 7.1 shows that super-resolutions with deblurring at the end beat those without deblurring in terms of pose estimation; however, super-resolutions without deblurring beat those with deblurring at the end in terms of the $L^N_2$ norms. We don’t know exactly what is the cause yet.

- Increasing the number of images passed eight for constructing super-resolution images doesn’t always cut down the $L^N_2$ norm because occlusion and lighting artifacts increase while little additional information is provided. For example, the fourth block in Table 7.1 shows that when the number of images increases from 8 to 16, the $L^N_2$ norms goes from 154.6 up to 166.8.

- Increasing the number of eigenvectors generally decreases the size of $L^N_2$ norm because each additional eigenvector has less significant contribution to the size of the $L_2$ norm. For example, the first block of Table 7.1 shows that when the number of eigenvectors goes from 16 to 32, the $L^N_2$ is cut by about 128 using cubic convolution and about 74 and 79 using super-resolution with and without
deblurring.

- Increasing the number of eigenvectors generally, but not always, increases the accuracy of pose estimation. For example, the second block of Table 7.1 shows that when the number of eigenvectors goes from 16 to 32, $PE$ goes from 33.5 degrees down to 0 degrees when using super-resolution, but it goes from 44.7 up to 47.5 degrees when using cubic convolution,

- Super-resolution does not always produce better results. In particular, we have found that if blurring the low-resolution images increases the accuracy of pose estimation, then super-resolution won’t help at all. The reason is that super-resolution goes in the opposite direction and attempts to improve the estimation of the high frequency components. For the object image set shown in Figure 7.4, our experimental results (Table 7.4) show that except at $0^\circ$ where super-resolution wins, blurring the low-resolution images before the construction of scaled-up images decreases the $L_2^N$ norms somewhere between 3.1% and 14.5% for both super-resolution and cubic convolution. At $270^\circ$ cubic-convolution’s pose error decreases 0.5 degrees, while at $90^\circ$ its pose error increases 1.7 degrees. This latter increase is odd, as the $L_2^N$ decreases at the same time. We believe that this is because the images, as shown in Figure 7.4, are dominated by undersampled or aliased high frequency text which changes radically with small perturbations. At $0^\circ$, super-resolution has better pose estimates, and the $L_2^N$ norms increase somewhere between 6.0% and 24.6% (with no change in pose estimation) for both super-resolution and cubic convolution.

- When super-resolution does not help, we found that keeping only low frequency
Table 7.5: Experimental results of object image set 1 using edge maps for both training and recognition.

<table>
<thead>
<tr>
<th>Object 1</th>
<th>16/128</th>
<th>32/128</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE</td>
<td>L_2^2</td>
<td>PE</td>
</tr>
<tr>
<td>X_{80}^-</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>CC_{m}</td>
<td>67.1</td>
<td>161.0</td>
</tr>
<tr>
<td>CC_{r}</td>
<td>0.0</td>
<td>0.2</td>
</tr>
<tr>
<td>SR_{8}</td>
<td>67.1</td>
<td>156.9</td>
</tr>
<tr>
<td>SR_{16}</td>
<td>67.1</td>
<td>156.0</td>
</tr>
<tr>
<td>SRD_{8}</td>
<td>67.1</td>
<td>155.6</td>
</tr>
<tr>
<td>SRD_{16}</td>
<td>67.1</td>
<td>166.1</td>
</tr>
<tr>
<td>SRD_{20}</td>
<td>67.1</td>
<td>165.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object 1</th>
<th>16/128</th>
<th>32/128</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE</td>
<td>L_2^2</td>
<td>PE</td>
</tr>
<tr>
<td>X_{180}^-</td>
<td>-1.1</td>
<td>0.0</td>
</tr>
<tr>
<td>CC_{m}</td>
<td>51.8</td>
<td>192.7</td>
</tr>
<tr>
<td>CC_{r}</td>
<td>109.9</td>
<td>-31.5</td>
</tr>
<tr>
<td>SR_{8}</td>
<td>114.6</td>
<td>212.4</td>
</tr>
<tr>
<td>SR_{16}</td>
<td>114.6</td>
<td>211.1</td>
</tr>
<tr>
<td>SRD_{8}</td>
<td>114.6</td>
<td>205.0</td>
</tr>
<tr>
<td>SRD_{16}</td>
<td>114.6</td>
<td>229.7</td>
</tr>
<tr>
<td>SRD_{20}</td>
<td>114.6</td>
<td>228.7</td>
</tr>
</tbody>
</table>

The acquired low-resolution images need to have enough high frequency information for image matching to work. Super-resolution relies heavily on good image matching results to bring the information together. If the result of image matching is poor, super-resolution probably won’t help.

- A more robust segmentation algorithm would help better separate the object region and the background region.
7.6 Conclusion

The fundamental contributions of this chapter can be summarized as follows: A generic approach is presented for quantitative measurement of super-resolution imaging using appearance matching and pose estimation. It can be used as a measure of any super-resolution algorithm—proposed or to-be proposed. As an example, this chapter showed the quantitative evaluation of the image-based super-resolution algorithm described in Chapter 4. On average, using 32 eigenvectors, image rescaling using cubic convolution had a pose error of 8.12 degrees while super-resolution with 8 images, deblurring or not, had a pose error of 1.4 degrees.
Chapter 8

Conclusions and Future Work

This chapter presents the conclusions and the future work, with the emphasis on issues that need to be addressed before the super-resolution algorithms described herein are robust enough for general use in applications.

8.1 Conclusions

Aimed at establishing a framework for applications that require sufficiently accurate warped intensity values to perform their tasks, this thesis first introduced a new class of reconstruction/restoration algorithms that is fundamentally different from traditional approaches. We deviate from the standard practice that treats images as “point” samples. In this work, image values are treated as area samples generated by non-overlapping integrators. This is consistent with the image formation process, particularly for CCD, CID, and almost any digital camera. We show that superior results are obtained by formulating reconstruction as a two-stage process: image restoration followed by application of the degradation model of the imaging sensor. By coupling the degradation model directly
into the reconstruction process, we satisfy a more intuitive fidelity measure of accuracy that is based on the physical limitations of the sensor. Efficient local techniques for image restoration are derived to invert the effects of the degradation model and estimate the underlying image that passed through the sensor. The reconstruction algorithms derived herein are local methods that compare favorably to cubic convolution, a well-known local technique, and they even rival global algorithms such as interpolating cubic splines. We show in this thesis that if linear interpolation is used to derive the value of the reconstruction at the pixel boundaries, the resulting imaging-consistent reconstruction algorithm QRR is tantamount to cubic convolution with the “optimal” value of $a = -0.5$. Evaluations are made by comparing their passband and stopband performances in the frequency domain, as well as by direct inspection of the resulting images in the spatial domain. A secondary advantage of the algorithms derived with this approach is that they satisfy an imaging-consistency property. This means that they exactly reconstruct the image for some function in the given class of functions. Their error can be shown to be at most twice that of the “optimal” algorithm for a wide range of optimality constraints.

We then generalized the idea of imaging-consistent reconstruction/restoration algorithms to deal with image warping. Whereas imaging-consistent reconstruction/restoration algorithms assume that the degradation model is the same for both input and output; imaging-consistent warping algorithms go one step further, allowing

1. both input and output to have their own degradation model, and
2. the degradation model to vary its size for each output pixel.

The imaging-consistent warping algorithms (also known as the integrating resamplers) provide an efficient method for image warping using the class of imaging-consistent
reconstruction/restoration algorithms. We show that by coupling the degradation model of the imaging system directly into the integrating resampler, we can better approximate the warping characteristics of real sensors, which also significantly improve the accuracy of the warped intensity values.

When we are resampling the image and warping its geometry in a nonlinear manner, this new approach allows us to efficiently do both pre-filtering and post-filtering. Because we have already determined a functional form for the input, no spatially-varying filtering is needed, as would be the case if direct inverse mapping were done. The integrating resampler described herein also handles antialiasing of partial pixels in a straightforward manner.

Examples are given to show the use of these imaging-consistent warping algorithms in two computer vision applications. The first is geometric correction for images degraded by radial lens distortion, demonstrated as a preprocessing step in correlation-based stereo matching. The second is warping-based data fusion for polarization computations in a small-baseline multi-camera system. These two applications are chosen because they exemplify the kinds of applications for which the imaging-consistent warping algorithms are designed—problems that use warped intensity values (as opposed to edge structure).

We then presented two new algorithms for enhancing image resolution from an image sequence. The “image-based” approach presumes that the images were taken under the same illumination conditions and uses the intensity information provided by the image sequence to construct the super-resolution image. When imaging from different viewpoints, over long temporal spans, or imaging scenes with moving 3D objects, the image intensities naturally vary. The “edge-based” approach, based on edge mod-
els and a local blur estimate, circumvents the difficulties caused by lighting variations. The approaches we propose herein use the imaging-consistent warping algorithms as the underlying resampling algorithm. We also generalize the idea of the imaging-consistent reconstruction/restoration algorithms to deal with discontinuities in an image. We show that the super-resolution problem may be solved by a direct method, which is fundamentally different from the iterative back-projection approaches proposed in the previous work. Results of our experiments show that the image-based approach described herein is not only computationally cheaper, but it also gives results comparable to or better than those using back-projection. Our experiments also show that image warping techniques may have a strong impact on the quality of image resolution enhancement, which have been completely ignored by earlier research on super-resolution. Moreover, we also demonstrate that super-resolution provides added benefits even if the final sampling rate is exactly the same as the original.

We then proposed two new algorithms to deal with the issue of quantitatively measuring the advantages of super-resolution algorithms, which has not addressed in previous work. The “OCR based” approach uses OCR as the fundamental measure. The “appearance matching and pose estimation based” approach uses appearance matching and pose estimation as the primary metric and image-quality metric as a secondary measure. We show that even when the images are qualitatively similar, quantitative differences appear in machine processing.
8.2 Future Work

In addition to edge localization and local blur estimation, the super-resolution algorithms described herein recognize four separate components: the matching (to determine alignment), the warping (to align the data and increase sampling rate), the fusion (to produce a less noisy image), and an optional deblurring stage (to remove lens blur). In this thesis, we have been concentrating our efforts on warping. Future work is needed before the super-resolution algorithms described herein are robust enough for general use in applications. In particular, the following issues need to be addressed, and more robust algorithms should be considered.

Motion estimation For motion estimation, we are using traditional matching on image fields (either sum-of-square difference or normalized correlation). The matching is sometimes inaccurate. We need to incorporate more robust sub-pixel matching algorithms, which have to be flexible enough to find more complex warps and to avoid lighting variation problem.

Fusion For fusion, we have experimented with simple averaging, averaging with trimmed tails, and median. These produce decreasingly accurate estimates with increasing robustness to outliers. As the matching is sometimes inaccurate and because of aliasing artifacts, a few outliers are common; thus, the trimmed tails is probably the best overall technique. However, more robust algorithms should be considered.

Deblurring For deblurring, we are using traditional deblurring techniques. The problem with these traditional techniques is that they cause ringing around the edges. We need to include more robust deblurring algorithms to reduce the undesirable effect.
3D edge matching For the edge-based super-resolution algorithm, more robust algorithms are needed to precisely match and fuse 3D edges symbolically under different viewing/lighting conditions, particularly, when more complex warps are involved.

Reestimation of the underlying image The idea of the edge-based super-resolution algorithm described in Chapter 5 is to use the edge/blur models to reestimate the underlying image. The reestimation algorithm described therein uses $v$ and $\delta$ to reestimate the underlying image. The possibility of incorporating $\sigma$ into the reestimation algorithm should be explored.

Quantitative measurement of super-resolution Due to the availability of the OCR programs, Chapter 6 shows the results of a commercial OCR program. If availability is not an issue, we should try different OCR programs.
Bibliography


[Chiang and Boult, 1995] Ming-Chao Chiang and Terrance E. Boult. A public domain system for camera calibration and distortion correction. Tech. Rep. CUCS-038-95,
Columbia University, Department of Computer Science, and Lehigh University, Department of EECS, Dec 1995.


Appendix A

Camera Calibration and Distortion Correction

The purpose of this chapter is threefold. First, we review a two-stage algorithm for camera calibration using off-the-shelf TV cameras and lenses [Tsai-1986, Tsai-1987, Chiang and Boult-1995]. This review is included primarily for its use in Chapter 3 and Chapter 6 to calibrate two cameras which have noticeable radial lens distortion. Second, we introduce an algorithm for automatically computing the image coordinates of the calibration points from the image of a calibration pattern. Third, we present an imaging-consistent algorithm for “unwarping” radial lens distortions once the underlying camera model is determined. Examples are given to show the use of these algorithms. Details of the implementation of the two-stage calibration algorithm and a set of utility programs—including programs that convert distorted image coordinates to undistorted image coordinates, 3D world coordinates to 2D image coordinates, 2D image coordinates to 3D world coordinates, and programs that generate synthetic data—can be found in [Chiang and Boult-1995].
A.1 Introduction

Camera calibration refers to the process of determining the internal and external parameters of a camera, for the following two purposes:

1. Inferring 3D information from computer image coordinates, and
2. Inferring 2D computer image coordinates from 3D information.

These two purposes are best served if the camera calibration technique is autonomous, efficient, versatile, and requires only common off-the-shelf cameras and lenses. The advantages of using off-the-shelf cameras and lenses are: versatility, availability, and familiarity. The two-stage algorithm reviewed herein is designed particularly for this purpose.

However, there are at least two questions to ask. First, given the image of a calibration pattern, how do we measure the image coordinates of the calibration points in the pattern image? Though it is always possible to measure these coordinates by hand, it is probably not an enjoyable work. Thus, we introduce an algorithm for automatically computing the image coordinates from a pattern image. Second, once the camera model is determined, how do we “undo” the effect of radial lens distortions? For many vision applications that rely vastly on the intensity values to perform their tasks, the precision of the warped intensity values is important. Thus, we present an algorithm for precisely unwarping the images degraded by radial lens distortions. This algorithm, designed particularly for this purpose, is imaging-consistent in the sense that was described in Chapter 3.

This chapter is organized as follows. The two-stage algorithm for camera calibration using off-the-shelf cameras and lenses is reviewed in Section A.2. The algorithm for
automatically computing the image coordinates of the calibration points from the image of a calibration pattern is presented in Section A.5. Section A.6 introduces an imaging-consistent algorithm for unwarping the images degraded by radial lens distortions.

A.2 Camera Calibration

In this section, we review the two-stage algorithm for camera calibration using off-the-shelf cameras and lenses. The algorithm described herein is “heavily” based on [Tsai-1986, Tsai-1987] because the implementations are. However, we are doing our best to make it more readable to those interested in calibrating a camera. In what follows, we first describe the underlying camera model and the definition of the parameters to be calibrated and then present the two-stage camera calibration algorithm and the theoretical derivation and other issues.

A.2.1 The Camera Model

This section describes the camera model, defines the parameters to be calibrated, and presents the simple parallelism principle.

A.2.1.1 The Four Steps Transformation from 3D World Coordinate System to 3D Camera Coordinate System

Figure A.1 illustrates the geometry of the camera model. \((x_w, y_w, z_w)\) is the 3D coordinate of the object point \(P\) in the 3D world coordinate system. \((x, y, z)\) is the 3D coordinate of the object point \(P\) in the 3D camera coordinate system, which is centered at point \(o\), the optical center, with the \(z\) axis being the same as the optical axis. \((X, Y)\)
is the 2D sensor coordinate system centered at $O$ and parallel to the $x$ and $y$ axes. $f$ is the distance between front image plane and the optical center. $(X_u, Y_u)$ is the undistorted, or ideal, image coordinate of $(x, y, z)$ if a perfect pin-hole lens is used. $(X_d, Y_d)$ is the distorted, or true, image coordinate which differs from $(X_u, Y_u)$ due to radial lens distortion.

![Camera geometry with perspective projection and radial lens distortion.](image)

Figure A.1: Camera geometry with perspective projection and radial lens distortion.

However, since the unit for 2D image coordinate $(X_f, Y_f)$ in the computer frame memory is the number of pixels, which is different from the 2D sensor coordinate $(X, Y)$, additional parameters need to be specified and calibrated that relate the 2D sensor coordinate system in the front image plane to the 2D image coordinate system in the computer frame memory. The overall transformation from $(x_w, y_w, z_w)$ to $(X_f, Y_f)$ is depicted in Figure A.2. The following is the transformation in analytic form for the four steps in
Figure A.2.

\[ \begin{align*}
\text{3D world coordinate} & \ (x_w, y_w, z_w) \\
\downarrow & \\
\text{Step 1} & \\
\text{Rigid body transformation from} \ (x_w, y_w, z_w) \ \text{to} \ (x, y, z) \\
\text{Parameters to be calibrated:} & \ R \ \text{and} \ T \\
\downarrow & \\
3D \text{camera coordinate} & \ (x, y, z) \\
\downarrow & \\
\text{Step 2} & \\
\text{Perspective Projection with pin-hole lens geometry} \\
\text{Parameters to be calibrated:} & \ f \\
\downarrow & \\
\text{Undistorted image coordinate} & \ (X_u, Y_u) \\
\downarrow & \\
\text{Step 3} & \\
\text{Radial lens distortion} \\
\text{Parameters to be calibrated:} & \ k_1 \\
\downarrow & \\
\text{Distorted image coordinate} & \ (X_d, Y_d) \\
\downarrow & \\
\text{Step 4} & \\
\text{TV camera scanning and acquisition timing error} \\
\text{Parameters to be calibrated:} & \ s_x \ \text{and} \ (C_x, C_y) \\
\downarrow & \\
\text{Image coordinate in computer frame memory} & \ (X_f, Y_f)
\end{align*} \]

Figure A.2: The transformation from 3D world coordinate to 2D image coordinate.

**Step 1** Rigid body transformation from 3D world coordinate \((x_w, y_w, z_w)\) to 3D camera coordinate \((x, y, z)\)

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = R \begin{bmatrix}
  x_w \\
  y_w \\
  z_w
\end{bmatrix} + T
\]  

(A.1)
where $R$ and $T$ are, respectively, the rotation matrix and the translation vector

\[
R \equiv \begin{bmatrix}
r_1 & r_2 & r_3 \\
r_4 & r_5 & r_6 \\
r_7 & r_8 & r_9 \\
\end{bmatrix} \quad \text{and} \quad T \equiv \begin{bmatrix}
T_x \\
T_y \\
T_z \\
\end{bmatrix}
\]  

(A.2)

Parameters to be calibrated: $R$ and $T$

**Step 2** Transformation from 3D camera coordinate $(x, y, z)$ to undistorted image coordinate $(X_u, Y_u)$ using perspective projection with pin-hole lens geometry

\[
X_u = f \frac{x}{z} \quad \text{and} \quad Y_u = f \frac{y}{z}
\]  

(A.3)

Parameters to be calibrated: effective focal length $f$

**Step 3** Radial lens distortion

\[
X_d + D_x = X_u \quad \text{and} \quad Y_d + D_y = Y_u
\]  

(A.4)

\[
D_x = X_d(\kappa_1 r^2), \quad D_y = Y_d(\kappa_1 r^2), \quad \text{and} \quad r = \sqrt{X_d^2 + Y_d^2}
\]  

(A.5)

where $(X_d, Y_d)$ is the distorted image coordinate on the image plane.

Parameters to be calibrated: first radial lens distortion coefficient $\kappa_1$

**Step 4** Distorted image coordinate $(X_d, Y_d)$ to computer image coordinate $(X_f, Y_f)$ transformation

\[
X_f = s_x d_x^{-1} X_d + C_x, \quad Y_f = d_y^{-1} Y_d + C_y
\]  

(A.6)

\[
d_x^d = d_x N_{ex} / N_{fx}
\]  

(A.7)

where
\((X_f, Y_f)\): image coordinate in computer frame memory

\((C_x, C_y)\): computer image coordinate for origin in image plane

\(d_x\): distance between adjacent sensor elements in \(X\) direction

\(d_y\): distance between adjacent sensor elements in \(Y\) direction

\(N_{cx}\): number of sensor elements in camera’s \(X\) direction

\(N_{fx}\): number of pixels in frame grabber’s \(X\) direction

Parameters to be calibrated: uncertainty image scale factor \(s_x\) and the image origin \((C_x, C_y)\).

A.2.1.2 Equations Relating the 3D World Coordinate to the 2D Image Coordinate

By combining the last three steps, the 2D image coordinate \((X, Y)\) is related to the 3D camera coordinate of the object point \(P\) by the following equations.

\[
s^{-1}_x d'_x X + s^{-1}_x d'_x X (\kappa_1 r^2) = \frac{f_x}{z}
\]

(A.8)

\[
d_y Y + d_y Y (\kappa_1 r^2) = \frac{f_y}{z}
\]

(A.9)

Substituting Eq. (A.1) into Eq. (A.8) and Eq. (A.9) gives

\[
s^{-1}_x d'_x X + s^{-1}_x d'_x X (\kappa_1 r^2) = \frac{r_1 x_w + r_2 y_w + r_3 z_w + T_x}{r_7 x_w + r_8 y_w + r_9 z_w + T_z}
\]

(A.10)

\[
d_y Y + d_y Y (\kappa_1 r^2) = \frac{r_4 x_w + r_5 y_w + r_6 z_w + T_y}{r_7 x_w + r_8 y_w + r_9 z_w + T_z}
\]

(A.11)

where \(X = X_f - C_x\), \(Y = Y_f - C_y\), and \(r = \sqrt{(s^{-1}_x d'_x X)^2 + (d_y Y)^2}\).
A.2.2 Parallelism Constraints

The idea of the two-stage camera calibration algorithm is based on the following four observations, as illustrated in Figure A.3.

**Observation 1** No matter how much the radial lens distortion is, the direction of the vector $\overrightarrow{OP_d}$ extending from the origin $O$ of the image plane to the image point $(X_d, Y_d)$ remains unchanged and is parallel to the vector $\overrightarrow{OzP}$ extending from the optical axis to the object point $(x, y, z)$.

**Observation 2** The effective focal length does not influence the direction of the vector $\overrightarrow{OP_d}$, for $f$ scales the image coordinate $X_d$ and $Y_d$ by the same rate.
Observation 3  Once the object world coordinate system is rotated and translated in \(x\) and \(y\) as in Step 1 in Figure A.2 such that \(OP_d\) is parallel to \(O_zP\) for every point, transformation in \(z\) will not alter the direction of \(OP_d\).

Observation 4  The constraint that \(OP_d\) is parallel to \(O_zP\) for every point, being independent of the radial lens distortion coefficient \(\kappa_1\), the effective focal length \(f\), and the \(z\) component of the translation vector \(T\), is actually sufficient to determine the 3D rotation matrix, the \(x\) and \(y\) components of the 3D translation vector, and the uncertainty scale factor \(s_x\).

A.2.3  Problem Definition

The problem of camera calibration is to compute the camera internal and external parameters based on a number of points in the calibration pattern whose world coordinates \((x_w, y_w, z_w)\) in the world coordinate system are known, and whose image coordinates \((X_f, Y_f)\) in the image coordinate system are measured or computed. See Section A.5 for details. The parameters to be calibrated can be categorized into the following two classes.

A.2.3.1  External Parameters

The parameters in Step 1 in Figure A.2 for the transformation from the object 3D world coordinate system to the camera 3D coordinate system centered at the optical center are called the external, or extrinsic, parameters. There are six external parameters: the direction angles yaw \(\theta\), pitch \(\phi\), and roll \(\psi\), and the three components of the translation
vector $T$. The rotation matrix can be expressed as a function of $\theta$, $\phi$, and $\psi$, as follows:

$$
R = \begin{bmatrix}
\cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi \\
\sin \psi \cos \theta & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi \\
-\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi
\end{bmatrix}
$$

(A.12)

### A.2.3.2 Internal Parameters

The parameters in Step 2, 3, and 4 in Figure A.2 for the transformation from the 3D camera coordinate system to the 2D image coordinate system are called internal, or intrinsic, parameters. There are five internal parameters:

- $f$: effective focal length
- $\kappa_1$: first radial lens distortion coefficient
- $s_x$: uncertainty scale factor for $x$, due to TV camera scanning and acquisition timing error
- $(C_x, C_y)$: the origin of the image coordinate in the image plane

### A.2.4 Calibrating Camera Using a Single View of Coplanar Points

This section describes camera calibration using a single view of a set of coplanar points [Tsai-1986, Tsai-1987]. Figure A.4 illustrates the experimental setup.

#### A.2.4.1 Stage 1: Compute $R$, $T_x$, and $T_y$

(i) *Compute the distorted image coordinates $(X_d, Y_d)$*
1. Grab a frame of the calibration pattern into the computer frame memory.

2. Compute the coordinate \((X_i^f, Y_i^f)\) of each calibration point \(i\). See Section A.5 for details.

3. Obtain \(N_{cx}, N_{fx}, d_x^f\), and \(d_y\) using information of camera provided by the
manufacturer.

4. Compute \((X_d^i, Y_d^i)\) using Eq. (A.6):

\[
X_d^i = s_x^{-1} d_x^i (X_f^i - C_x) \quad Y_d^i = d_y (Y_f^i - C_y)
\]

for \(i = 1, \ldots, N\) where \(N\) is the total number of calibration points.

(ii) Compute \(r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y,\) and \(r_5/T_y\)

For each calibration point \(i\) with \((x_w^i, y_w^i, z_w^i)\) as its 3D world coordinate and \((X_d^i, Y_d^i)\) as the corresponding 2D image coordinate (computed in (i) above), set up the following system of linear equations with \(r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y,\) and \(r_5/T_y\) as unknowns:

\[
Ax = b \tag{A.13}
\]

where

\[
A = \begin{bmatrix}
Y_d^1 x_w^1 & Y_d^1 y_w^1 & Y_d^1 & -X_d^1 x_w^1 & -X_d^1 y_w^1 \\
Y_d^2 x_w^2 & Y_d^2 y_w^2 & Y_d^2 & -X_d^2 x_w^2 & -X_d^2 y_w^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
Y_d^N x_w^N & Y_d^N y_w^N & Y_d^N & -X_d^N x_w^N & -X_d^N y_w^N
\end{bmatrix},
\]

\[
x = \begin{bmatrix}
r_1/T_y \\
r_2/T_y \\
T_x/T_y \\
r_4/T_y \\
r_5/T_y
\end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix}
X_d^1 \\
X_d^2 \\
\vdots \\
X_d^N
\end{bmatrix}.
\]

With \(N\) much larger than five, an overdetermined system of linear equations can be established and solved for the five unknowns \(r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y,\) and \(r_5/T_y\), in the least squares sense.
(iii) Compute $R$, $T_x$, and $T_y$

(iii.1) Compute $|T_y|$ from $r_1/T_y$, $r_2/T_y$, $T_x/T_y$, $r_4/T_y$, and $r_5/T_y$.

Let $C$ be a 2x2 submatrix of the rotation matrix $R$; that is, $C$ is defined as

$$
C \equiv \begin{bmatrix}
    r'_1 & r'_2 \\
    r'_4 & r'_5
\end{bmatrix} \equiv \begin{bmatrix}
    r_1/T_y & r_2/T_y \\
    r_4/T_y & r_5/T_y
\end{bmatrix}
$$

(A.14)

\text{IF not} a whole row or column of $C$ vanishes

THEN compute $T^2_y$ with

$$
T^2_y = \frac{S_r - \sqrt{S_r^2 - 4(r'^2_1 r'^2_5 - r'^2_4 r'^2_2)}}{2(r'^2_1 r'^2_5 - r'^2_4 r'^2_2)}
$$

(A.15)

where $S_r = r'^2_1 + r'^2_2 + r'^2_4 + r'^2_5$.

ELSE compute $T^2_y$ with

$$
T^2_y = (r'^2_i + r'^2_j)^{-1}
$$

(A.16)

where $r'^i_i$ and $r'^j_j$ are the elements in the row or column of $C$ that do not vanish. This rarely happens, if ever.

(iii.2) Determine the sign of $T_y$

1. Pick a calibration point $i$ whose world coordinate is $(x^i_w, y^i_w, z^i_w)$ and whose image coordinate $(X^i_f, Y^i_f)$ is away from the image center $(C_x, C_y)$.

2. Pick the sign of $T_y$ to be $+1$.

3. Compute the following:

$$
r_1 = (r_1/T_y) \cdot T_y
$$

$$
r_2 = (r_2/T_y) \cdot T_y
$$
\[ T_x = (T_x/T_y) \cdot T_y \]
\[ r_1 = (r_1/T_y) \cdot T_y \]
\[ r_2 = (r_2/T_y) \cdot T_y \]
\[ T_x = (T_x/T_y) \cdot T_y \]
\[ r_4 = (r_4/T_y) \cdot T_y \]
\[ r_5 = (r_5/T_y) \cdot T_y \]

\[ x = r_1 x_w + r_2 y_w + r_3 \cdot 0 + T_x \]
\[ y = r_4 x_w + r_5 y_w + r_6 \cdot 0 + T_y \]

where \( r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y, \) and \( r_5/T_y \) are determined in (ii).

4. **IF** ((\( x \) and \( X \) have the same sign) and (\( y \) and \( Y \) have the same sign))

    **THEN** \( \text{sign}(T_y) = +1 \)

    **ELSE** \( \text{sign}(T_y) = -1 \)

(iii.3) *Compute the rotation matrix \( R \)*

1. Compute the following:

\[ r_1 = (r_1/T_y) \cdot T_y \]
\[ r_2 = (r_2/T_y) \cdot T_y \]
\[ T_x = (T_x/T_y) \cdot T_y \]
\[ r_4 = (r_4/T_y) \cdot T_y \]
\[ r_5 = (r_5/T_y) \cdot T_y \]

where \( r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y, \) and \( r_5/T_y \) are determined in (ii).

2. Compute the effective focal length \( f \) using Eq. (A.19)

    **IF** \( (f \geq 0) \)
THEN

\[
R = \begin{bmatrix}
    r_1 & r_2 & \sqrt{(1 - r_1^2 - r_2^2)} \\
    r_4 & r_5 & s \sqrt{(1 - r_4^2 - r_5^2)} \\
    r_7 & r_8 & r_9
\end{bmatrix}
\]  \hspace{1cm} (A.17)

ELSE

\[
R = \begin{bmatrix}
    r_1 & r_2 & -\sqrt{(1 - r_1^2 - r_2^2)} \\
    r_4 & r_5 & -s \sqrt{(1 - r_4^2 - r_5^2)} \\
    -r_7 & -r_8 & r_9
\end{bmatrix}
\]  \hspace{1cm} (A.18)

where \( s = -\text{sign}(r_1r_4 + r_2r_5) \), and \( r_7, r_8, \) and \( r_9 \) are determined from the cross product of the first two rows using the orthonormal and right handed property of \( R \).

A.2.4.2 Stage 2: Compute Effective Focal Length \( f \), First Radial Lens Distortion Coefficient \( \kappa_1 \), and \( T_z \)

(iv) Compute an approximation of \( f \) and \( T_z \) by ignoring radial lens distortion

For each calibration point \( i \), establish the system of linear equations with \( f \) and \( T_z \) as unknowns, as follows:

\[
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N
\end{bmatrix}
- \begin{bmatrix}
d_y Y_1 \\
d_y Y_2 \\
\vdots \\
d_y Y_N
\end{bmatrix}
= \begin{bmatrix}
w_1 d_y Y_1 \\
w_2 d_y Y_2 \\
\vdots \\
w_N d_y Y_N
\end{bmatrix}
\]  \hspace{1cm} (A.19)

where \( y_i = r_4x_i^w + r_5y_i^w + r_6 \cdot 0 + T_y \) and \( w_i = r_7x_i^w + r_8y_i^w + r_9 \cdot 0 \).

With \( N \) calibration points, this yields an overdetermined system of linear equations that can be solved for the unknowns \( f \) and \( T_z \).
Compute the exact solution for $f, T_z,$ and $\kappa_1$

Solve Eq. (A.11) with $f, T_z,$ and $\kappa_1$ as unknowns using nonlinear optimization scheme such as steepest decent. Use the approximation for $f$ and $T_z$ computed in (iv) as the initial guess for $f$ and $T_z$ and zero as the initial guess for $\kappa_1$.

A.2.5 Calibrating Camera Using a Single View of Non-Coplanar Points

This section describes camera calibration using a single view of a set of non-coplanar points [Tsai-1986, Tsai-1987]. Figure A.5 illustrates the experimental setup. This is different from the coplanar case in that the same calibration pattern is translated to several different heights. Note that some of the materials described in this section are intentionally repeated from the previous section to make it more self-contained.

A.2.5.1 Stage 1: Compute $R, T_x, T_y,$ and Uncertainty Scale Factor $s_x$

(i) Compute the distorted image coordinates $(X_d, Y_d)$, where $(X_d, Y_d)$ is defined exactly the same as the $(X_f, Y_f)$ in A.6 except that $s_x$ is set to 1.

1. Grab a frame of the calibration pattern into the computer frame memory.

2. Compute the coordinate $(X_i^j, Y_i^j)$ of each calibration point $i$. See Section A.5 for details.

3. Obtain $N_{cx}, N_{fx}, d_x^f,$ and $d_y$ using information of camera provided by the manufacturer.

4. Compute $(X_d^i, Y_d^i)$ using Eq. (A.6):

$$X_d^i = s_x^{-1} d_x^f (X_f^i - C_x) \quad Y_d^i = d_y (Y_f^i - C_y)$$
Figure A.5: Illustration of experimental setup for camera calibration using a set of non-coplanar points.

\[(ii) \text{ Compute } s_{x1}/T_y, s_{x2}/T_y, s_{x3}/T_y, s_x/T_y, r_4/T_y, r_5/T_y, \text{ and } r_6/T_y\]

For each calibration point \(i\) with \((x^i_w, y^i_w, z^i_w)\) as its 3D world coordinate and \((\overline{X}_d, \overline{Y}_d)\) as the corresponding 2D image coordinate (computed in (i) above), set up the
following system of linear equations with \( s_{x r_1}/T_y, s_{x r_2}/T_y, s_{x r_3}/T_y, s_x T_x/T_y, r_4/T_y, r_5/T_y, \) and \( r_6/T_y \) as unknowns:

\[
Ax = b
\]  \hspace{1cm} (A.20)

where

\[
A \equiv \begin{bmatrix}
\gamma^1_{dxw} & \gamma^1_{dyw} & \gamma^1_{dzw} & \gamma^1_d & -\gamma^1_{dxw} & -\gamma^1_{dyw} & -\gamma^1_{dzw} \\
\gamma^2_{dxw} & \gamma^2_{dyw} & \gamma^2_{dzw} & \gamma^2_d & -\gamma^2_{dxw} & -\gamma^2_{dyw} & -\gamma^2_{dzw} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\gamma^N_{dxw} & \gamma^N_{dyw} & \gamma^N_{dzw} & \gamma^N_d & -\gamma^N_{dxw} & -\gamma^N_{dyw} & -\gamma^N_{dzw}
\end{bmatrix},
\]

\[
x \equiv \begin{bmatrix}
s_{x r_1}/T_y \\
s_{x r_2}/T_y \\
s_{x r_3}/T_y \\
s_x T_x/T_y \\
r_4/T_y \\
r_5/T_y \\
r_6/T_y
\end{bmatrix}, \quad \text{and} \quad b \equiv \begin{bmatrix}
\gamma^1_d \\
\gamma^2_d \\
\vdots \\
\gamma^N_d
\end{bmatrix}.
\]

With \( N \) much larger than seven, an overdetermined system of linear equations can be established and solved for the five unknowns \( s_{x r_1}/T_y, s_{x r_2}/T_y, s_{x r_3}/T_y, s_x T_x/T_y, r_4/T_y, r_5/T_y, \) and \( r_6/T_y \), in the least squares sense.

(iii) Compute \( R, T_x, \) and \( T_y \)

For simplicity of discussion, let \( a_i, i = 1, \ldots, 7 \) be defined as \( a_1 = s_{x r_1}/T_y, a_2 = s_{x r_2}/T_y, a_3 = s_{x r_3}/T_y, a_4 = s_x T_x/T_y, a_5 = r_4/T_y, a_6 = r_5/T_y, \) and \( a_7 = r_6/T_y \). Note that all the \( a_i, i = 1, \ldots, 7 \) are determined in (ii).
(iii.1) Compute $|T_y|$ from $s_x r_1 / T_y$, $s_x r_2 / T_y$, $s_x r_3 / T_y$, $s_x T_x / T_y$, $r_4 / T_y$, $r_5 / T_y$, and $r_6 / T_y$

Compute $|T_y|$ using the following formula:

$$|T_y| = 1 / \sqrt{a_5^2 + a_6^2 + a_7^2}$$ (A.21)

(iii.2) Determine the sign of $T_y$

1. Pick a calibration point $i$ whose world coordinate is $(x^i_w, y^i_w, z^i_w)$ and whose image coordinate $(X^i_f, Y^i_f)$ is away from the image center $(C_x, C_y)$.

2. Pick the sign of $T_y$ to be +1.

3. Compute the following:

$$r_1 = (r_1 / T_y) \cdot T_y$$
$$r_2 = (r_2 / T_y) \cdot T_y$$
$$r_3 = (r_3 / T_y) \cdot T_y$$
$$T_x = (T_x / T_y) \cdot T_y$$
$$r_4 = (r_4 / T_y) \cdot T_y$$
$$r_5 = (r_5 / T_y) \cdot T_y$$
$$r_6 = (r_6 / T_y) \cdot T_y$$

$$x = r_1 x_w + r_2 y_w + r_3 z_w + T_x$$
$$y = r_4 x_w + r_5 y_w + r_6 z_w + T_y$$

where $s_x r_1 / T_y$, $s_x r_2 / T_y$, $s_x r_3 / T_y$, $s_x T_x / T_y$, $r_4 / T_y$, $r_5 / T_y$, and $r_6 / T_y$ are determined in (ii).
4. **IF** ((x and X have the same sign) and (y and Y have the same sign))

**THEN** \( \text{sign}(T_y) = +1 \)

**ELSE** \( \text{sign}(T_y) = -1 \)

(iii.3) **Determine** \( s_x \)

Use the following formula to compute \( s_x \):

\[
s_x = (a_1^2 + a_2^2 + a_3^2)^{1/2}[T_y]
\]

(A.22)

(iii.4) **Compute the rotation matrix** \( R \)

Compute \( r_i, i = 1, \ldots, 6 \), and \( T_x \) with the following formula:

\[
\begin{align*}
    r_1 & = a_1 \cdot T_y / s_x \\
    r_2 & = a_2 \cdot T_y / s_x \\
    r_3 & = a_3 \cdot T_y / s_x \\
    T_x & = a_4 \cdot T_y / s_x \\
    r_5 & = a_5 \cdot T_y \\
    r_6 & = a_6 \cdot T_y \\
    r_7 & = a_7 \cdot T_y
\end{align*}
\]

where \( a_i, i = 1, \ldots, 7 \) are determined in (iii).

Given \( r_i, i = 1, \ldots, 6 \), which are the elements in the first two rows of \( R \), the third row of \( R \) can be computed as the cross product of the first two rows, using the orthonormal property of \( R \) and the right handed rule (determinant of \( R = 1 \), not \(-1\)).
A.2.5.2 Stage 2: Compute Effective Focal Length $f$, First Radial Lens Distortion Coefficient $\kappa_1$, and $T_z$

(iv) Compute an approximation of $f$ and $T_z$ by ignoring radial lens distortion

For each calibration point $i$, establish the system of linear equations with $f$ and $T_z$ as unknowns, as follows:

$$
\begin{bmatrix}
y_1 & -d_y Y_1 \\
y_2 & -d_y Y_2 \\
\vdots & \vdots \\
y_N & -d_y Y_N
\end{bmatrix}
\begin{bmatrix}
f \\
T_z
\end{bmatrix}
= 
\begin{bmatrix}
w_1 d_y Y_1 \\
w_2 d_y Y_2 \\
\vdots \\
w_N d_y Y_N
\end{bmatrix}
$$

(A.23)

where $y_i = r_4 x_i^w + r_5 y_i^w + r_6 z_i^w + T_y$ and $w_i = r_7 x_i^w + r_8 y_i^w + r_9 z_i^w$.

With $N$ calibration points, this yields an overdetermined system of linear equations that can be solved for the unknowns $f$ and $T_z$.

(v) Compute the exact solution for $f$, $T_z$, and $\kappa_1$

Solve Eq. (A.11) with $f$, $T_z$, and $\kappa_1$ as unknowns using nonlinear optimization scheme such as steepest decent. Use the approximation for $f$ and $T_z$ computed in (iv) as the initial guess for $f$ and $T_z$ and zero as the initial guess for $\kappa_1$.

A.2.6 Calibrating External Parameters Using a Single View of Coplanar Points

This section describes a variant of the first stage of the two-stage camera calibration algorithm we described in Sections A.2.4 and A.2.5. This variant assumes that the camera’s internal parameters are known and calibrates only the external parameters using a single view of a coplanar set of points.
(i) **Compute the undistorted image coordinates** \((X_u, Y_u)\)

1. Grab a frame of the calibration pattern into the computer frame memory.
2. Compute the coordinate \((X^i_f, Y^i_f)\) of each calibration point \(i\). See Section A.5 for details.
3. Obtain \(N_{cx}, N_{fx}, d^i_x,\) and \(d_y\) using information of camera provided by the manufacturer.
4. Compute \((X^i_d, Y^i_d)\) using Eq. (A.6):

\[
X^i_d = s^{-1}_x d^i_x (X^i_f - C_x) \quad \text{and} \quad Y^i_d = d_y (Y^i_f - C_y)
\]

and then compute \((X^i_u, Y^i_u)\) using Eq. (A.4):

\[
X^i_u = X^i_d \sqrt{X^i_d^2 + Y^i_d^2} \quad \text{and} \quad Y^i_u = Y^i_d \sqrt{X^i_d^2 + Y^i_d^2}
\]

for \(i = 1, \ldots, N\) where \(N\) is the total number of calibration points.

(ii) **Compute** \(r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y,\) and \(r_5/T_y\)

For each calibration point \(i\) with \((x^i_w, y^i_w, z^i_w)\) as its 3D world coordinate and \((X^i_u, Y^i_u)\) as the corresponding undistorted 2D image coordinate (computed in (i) above), set up the following system of linear equations with \(r_1/T_y, r_2/T_y, T_x/T_y, r_4/T_y,\) and \(r_5/T_y\) as unknowns:

\[
Ax = b \tag{A.24}
\]
where

\[
A \equiv \begin{bmatrix}
Y_{w}^{1} x_{w} & Y_{w}^{1} y_{w} & Y_{u}^{1} & -X_{w}^{1} x_{w} & -X_{w}^{1} y_{w} \\
Y_{w}^{2} x_{w} & Y_{w}^{2} y_{w} & Y_{u}^{2} & -X_{w}^{2} x_{w} & -X_{w}^{2} y_{w} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
Y_{u}^{N} x_{w} & Y_{u}^{N} y_{w} & Y_{u}^{N} & -X_{u}^{N} x_{w} & -X_{u}^{N} y_{w}
\end{bmatrix},
\]

\[
x \equiv \begin{bmatrix}
r_{1}/T_{y} \\
r_{2}/T_{y} \\
T_{x}/T_{y} \\
r_{4}/T_{y} \\
r_{5}/T_{y}
\end{bmatrix}, \quad \text{and} \quad b \equiv \begin{bmatrix}
X_{u}^{1} \\
X_{d}^{2} \\
\vdots \\
X_{u}^{N}
\end{bmatrix}.
\]

With \(N\) much larger than five, an overdetermined system of linear equations can be established and solved for the five unknowns \(r_{1}/T_{y}, r_{2}/T_{y}, T_{x}/T_{y}, r_{4}/T_{y}, \) and \(r_{5}/T_{y}\), in the least squares sense.

(iii) **Compute** \(R_{x}, R_{y}, R_{z}, T_{x}, \) and \(T_{y}\)

(iii.1) **Compute** \(|T_{y}|\) from \(r_{1}/T_{y}, r_{2}/T_{y}, T_{x}/T_{y}, r_{4}/T_{y}, \) and \(r_{5}/T_{y}\).

Let \(C\) be a 2x2 submatrix of the rotation matrix \(R\); that is, \(C\) is defined as

\[
C \equiv \begin{bmatrix}
r'_{1} & r'_{2} \\
r'_{4} & r'_{5}
\end{bmatrix} \equiv \begin{bmatrix}
r_{1}/T_{y} & r_{2}/T_{y} \\
r_{4}/T_{y} & r_{5}/T_{y}
\end{bmatrix}
\]

(A.25)

**IF not** a whole row or column of \(C\) vanishes

**THEN** compute \(T_{y}^{2}\) with

\[
T_{y}^{2} = \frac{S_{r} - \sqrt{S_{r}^{2} - 4(r'_{1} r'_{5} - r'_{4} r'_{2})^{2}}}{2(r'_{1} r'_{5} - r'_{4} r'_{2})^{2}}
\]

(A.26)

where \(S_{r} = r'_{1}^{2} + r'_{2}^{2} + r'_{4}^{2} + r'_{5}^{2}\).
ELSE compute $T_y^2$ with

$$T_y^2 = (r_i^2 + r_j^2)^{-1} \tag{A.27}$$

where $r_i^2$ and $r_j^2$ are the elements in the row or column of $C$ that do not vanish. This rarely happens, if ever.

(iii.2) Determine the sign of $T_y$

1. Pick a calibration point $i$ whose world coordinate is $(x_i^i, y_i^i, z_i^i)$ and whose image coordinate $(X_i^i, Y_i^i)$ is away from the image center $(C_x, C_y)$.

2. Pick the sign of $T_y$ to be $+1$.

3. Compute the following:

\[
\begin{align*}
    r_1 & = (r_1/T_y) \cdot T_y \\
    r_2 & = (r_2/T_y) \cdot T_y \\
    T_x & = (T_x/T_y) \cdot T_y \\
    r_4 & = (r_4/T_y) \cdot T_y \\
    r_5 & = (r_5/T_y) \cdot T_y \\
    x & = r_1 x_w + r_2 y_w + r_3 \cdot 0 + T_x \\
    y & = r_4 x_w + r_5 y_w + r_6 \cdot 0 + T_y
\end{align*}
\]

where $r_1/T_y$, $r_2/T_y$, $T_x/T_y$, $r_4/T_y$, $r_5/T_y$, and $r_5/T_y$ are determined in (ii).

4. **IF** ($(x$ and $X$ have the same sign) and ($y$ and $Y$ have the same sign))

**THEN**  \(\text{sign}(T_y) = +1\)

**ELSE**  \(\text{sign}(T_y) = -1\)
(iii.3) Compute the rotation matrix $R$

1. Compute the following:

$$
\begin{align*}
    r_1 &= (r_1/T_y) \cdot T_y \\
    r_2 &= (r_2/T_y) \cdot T_y \\
    T_x &= (T_x/T_y) \cdot T_y \\
    r_4 &= (r_4/T_y) \cdot T_y \\
    r_5 &= (r_5/T_y) \cdot T_y
\end{align*}
$$

where $r_1/T_y$, $r_2/T_y$, $T_x/T_y$, $r_4/T_y$, and $r_5/T_y$ are determined in (ii).

2. Compute the effective focal length $f$ using Eq. (A.19)

**IF** ($f \geq 0$)

**THEN**

$$
R = \begin{bmatrix}
    r_1 & r_2 & \sqrt{1 - r_1^2 - r_2^2} \\
    r_4 & r_5 & s \sqrt{1 - r_4^2 - r_5^2} \\
    r_7 & r_8 & r_9
\end{bmatrix}
$$

(A.28)

**ELSE**

$$
R = \begin{bmatrix}
    r_1 & r_2 & -\sqrt{1 - r_1^2 - r_2^2} \\
    r_4 & r_5 & -s \sqrt{1 - r_4^2 - r_5^2} \\
    -r_7 & -r_8 & r_9
\end{bmatrix}
$$

(A.29)

where $s = -\text{sign}(r_1 r_4 + r_2 r_5)$, and $r_7$, $r_8$, $r_9$ are determined from the cross product of the first two rows using the orthonormal and right handed property of $R$.

(iv) Compute the translation vector $T$
(iv.1) Compute untranslated camera coordinates

\[
\begin{bmatrix}
\hat{x} \\
\hat{y} \\
\hat{z}
\end{bmatrix} = R
\begin{bmatrix}
x_w \\
y_w \\
z_w
\end{bmatrix}
\]

(iv.2) Compute \((X_u, Y_u)\) as in (i)

(iv.3) Solve the following overdetermined system of linear equations with \(T_x\), \(T_y\), and \(T_z\) as unknowns:

\[
\begin{bmatrix}
f & 0 & -X_u^1 \\
f & 0 & -X_u^2 \\
\vdots & \vdots & \vdots \\
f & 0 & -X_u^N \\
0 & f & -Y_u^1 \\
0 & f & -Y_u^2 \\
\vdots & \vdots & \vdots \\
0 & f & -Y_u^N
\end{bmatrix}
\begin{bmatrix}
T_x \\
T_y \\
T_z
\end{bmatrix}
= \begin{bmatrix}
\hat{z}X_u^1 - \hat{x}f \\
\hat{z}X_u^2 - \hat{x}f \\
\vdots \\
\hat{z}X_u^N - \hat{x}f \\
\hat{z}Y_u^1 - \hat{y}f \\
\hat{z}Y_u^2 - \hat{y}f \\
\vdots \\
\hat{z}Y_u^N - \hat{y}f
\end{bmatrix}
\]

(v) Optimize \(R\) and \(T\)

Take as the initial guess \(R_x, R_y, R_z, T_x, T_y,\) and \(T_z\) computed in the previous steps and optimize the errors in the least squares sense using non-linear optimization.

A.2.7 Calibrating External Parameters Using a Single View of Non-Coplanar Points

This section describes a variant of the first stage of the two-stage camera calibration algorithm we described in Sections A.2.4 and A.2.5. This variant assumes that the camera’s
internal parameters are known and calibrates only the external parameters using a single view of a non-coplanar set of points. Note again that some of the materials described in this section are intentionally repeated from the previous section to make it more self-contained.

(i) **Compute the undistorted image coordinates** \((X_u, Y_u)\)

1. Grab a frame of the calibration pattern into the computer frame memory.
2. Compute the coordinate \((X_i^f, Y_i^f)\) of each calibration point \(i\). See Section A.5 for details.
3. Obtain \(N_{ex}, N_{fx}, d_x', d_y\) using information of camera provided by the manufacturer.
4. Compute \((X_i^d, Y_i^d)\) using Eq. (A.6):

\[
X_i^d = s_x^{-1} d_x' (X_i^f - C_x) \quad \text{and} \quad Y_i^d = d_y (Y_i^f - C_y)
\]

and then compute \((X_i^i, Y_i^i)\) using Eq. (A.4):

\[
X_i^i = X_i^d \sqrt{X_i^{d^2} + Y_i^{d^2}} \quad \text{and} \quad Y_i^i = Y_i^d \sqrt{X_i^{d^2} + Y_i^{d^2}}
\]

for \(i = 1, \ldots, N\) where \(N\) is the total number of calibration points.

(ii) **Compute** \(s_x r_1/T_y, s_x r_2/T_y, s_x r_3/T_y, s_x T_x/T_y, r_4/T_y, r_5/T_y, \text{and } r_6/T_y\)

For each calibration point \(i\) with \((x_{w_i}^i, y_{w_i}^i, z_{w_i}^i)\) as its 3D world coordinate and \((X_d^i, Y_d^i)\) as the corresponding undistorted 2D image coordinate (computed in (i) above), set up the following system of linear equations with \(s_x r_1/T_y, s_x r_2/T_y, s_x r_3/T_y, s_x T_x/T_y, r_4/T_y, r_5/T_y, \text{and } r_6/T_y\) as unknowns:
\[ Ax = b \]  

where

\[
A \equiv \begin{bmatrix}
Y_u^1 x_w^1 & Y_u^1 y_w^1 & Y_u^1 z_w^1 & Y_u^1 & -X_u^1 x_w^1 & -X_u^1 y_w^1 & -X_u^1 z_w^1 \\
Y_u^2 x_w^2 & Y_u^2 y_w^2 & Y_u^2 z_w^2 & Y_u^2 & -X_u^2 x_w^2 & -X_u^2 y_w^2 & -X_u^2 z_w^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
Y_u^N x_w^N & Y_u^N y_w^N & Y_u^N z_w^N & Y_u^N & -X_u^N x_w^N & -X_u^N y_w^N & -X_u^N z_w^N
\end{bmatrix},
\]

\[
x = \begin{bmatrix}
s_{x r_1} / T_y \\
s_{x r_2} / T_y \\
s_{x r_3} / T_y \\
s_x T_x / T_y \\
r_4 / T_y \\
r_5 / T_y \\
r_6 / T_y
\end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix}
X_u^1 \\
X_u^2 \\
\vdots \\
X_u^N
\end{bmatrix}.
\]

With \( N \) much larger than seven, an overdetermined system of linear equations can be established and solved for the five unknowns \( s_{x r_1} / T_y, s_{x r_2} / T_y, s_{x r_3} / T_y, s_x T_x / T_y, r_4 / T_y, r_5 / T_y, \) and \( r_6 / T_y \), in the least squares sense.

(iii) **Compute the rotation matrix** \( R, T_x, \) and \( T_y \)

For simplicity of discussion, let \( a_i, i = 1, \ldots, 7 \) be defined as \( a_1 = s_{x r_1} / T_y, \)
\[
a_2 = s_{x r_2} / T_y, \ a_3 = s_{x r_3} / T_y, \ a_4 = s_x T_x / T_y, \ a_5 = r_4 / T_y, \ a_6 = r_5 / T_y, \text{ and } a_7 = r_6 / T_y. \]

Note that all the \( a_i, i = 1, \ldots, 7 \) are determined in (ii).

(iii.1) **Compute** \( |T_y| \) **from** \( s_{x r_1} / T_y, s_{x r_2} / T_y, s_{x r_3} / T_y, s_x T_x / T_y, r_4 / T_y, r_5 / T_y, \) and \( r_6 / T_y \)
Compute \(|T_y|\) using the following formula:

\[ |T_y| = 1/\sqrt{a_5^2 + a_6^2 + a_7^2} \]  \hspace{1cm} (A.31)

(iii.2) **Determine the sign of \(T_y\)**

1. Pick a calibration point \(i\) whose world coordinate is \((x_w^i, y_w^i, z_w^i)\) and whose image coordinate \((X^i_f, Y^i_f)\) is away from the image center \((C_x, C_y)\).

2. Pick the sign of \(T_y\) to be +1.

3. Compute the following:

\[
egin{align*}
    r_1 &= (r_1/T_y) \cdot T_y \\
    r_2 &= (r_2/T_y) \cdot T_y \\
    r_3 &= (r_3/T_y) \cdot T_y \\
    T_x &= (T_x/T_y) \cdot T_y \\
    r_4 &= (r_4/T_y) \cdot T_y \\
    r_5 &= (r_5/T_y) \cdot T_y \\
    r_6 &= (r_6/T_y) \cdot T_y \\
    x &= r_1 x_w + r_2 y_w + r_3 \cdot 0 + T_x \\
    y &= r_4 x_w + r_5 y_w + r_6 \cdot 0 + T_y
\end{align*}
\]

where \(s_x r_1/T_y, s_x r_2/T_y, s_x r_3/T_y, s_x T_x/T_y, r_4/T_y, r_5/T_y, \) and \(r_6/T_y\) are determined in (ii).

4. **IF** \((x \text{ and } X \text{ have the same sign}) \text{ and } (y \text{ and } Y \text{ have the same sign})\)

   **THEN** \(\text{sign}(T_y) = +1\)

   **ELSE** \(\text{sign}(T_y) = -1\)
(iii.3) Compute the rotation matrix $R$

Compute $r_i, i = 1, \ldots, 6$, and $T_x$ with the following formula:

\[
\begin{align*}
    r_1 &= a_1 \cdot T_y / s_x \\
    r_2 &= a_2 \cdot T_y / s_x \\
    r_3 &= a_3 \cdot T_y / s_x \\
    T_x &= a_4 \cdot T_y / s_x \\
    r_5 &= a_5 \cdot T_y \\
    r_6 &= a_6 \cdot T_y \\
    r_7 &= a_7 \cdot T_y
\end{align*}
\]

where $a_i, i = 1, \ldots, 7$ are determined in (iii).

Given $r_i, i = 1, \ldots, 6$, which are the elements in the first two rows of $R$, the third row of $R$ can be computed as the cross product of the first two rows, using the orthonormal property of $R$ and the right handed rule (determinant of $R = 1$, not $-1$).

(iv) Compute the translation vector $T$

(iv.1) Compute untranslated camera coordinates

\[
\begin{bmatrix}
    \hat{x} \\
    \hat{y} \\
    \hat{z}
\end{bmatrix} = R 
\begin{bmatrix}
    x_w \\
    y_w \\
    z_w
\end{bmatrix}
\]

(iv.2) Compute $(X_u, Y_u)$ as in (i)

(iv.3) Set up the following system of linear equations with $T_x$, $T_y$, and $T_z$ as unknowns:
\[
\begin{bmatrix}
  f & 0 & -X_u^1 \\
  f & 0 & -X_u^2 \\
  \vdots & \vdots & \vdots \\
  f & 0 & -X_u^N \\
  0 & f & -Y_u^1 \\
  0 & f & -Y_u^2 \\
  \vdots & \vdots & \vdots \\
  0 & f & -Y_u^N
\end{bmatrix}
\begin{bmatrix}
  T_x \\
  T_y \\
  T_z
\end{bmatrix}
= \begin{bmatrix}
  \hat{X}_u^1 - \hat{\alpha}f \\
  \hat{X}_u^2 - \hat{\alpha}f \\
  \vdots \\
  \hat{X}_u^N - \hat{\alpha}f \\
  \hat{Y}_u^1 - \hat{\gamma}f \\
  \hat{Y}_u^2 - \hat{\gamma}f \\
  \vdots \\
  \hat{Y}_u^N - \hat{\gamma}f
\end{bmatrix}
\]

(v) **Optimize R and T**

Take as the initial guess \( R_x, R_y, R_z, T_x, T_y, \) and \( T_z \) computed in the previous steps and optimize the errors in the least squares sense using non-linear optimization.

### A.3 Close Form Solution for Cubic Equations

The purpose of this section is to present a close form solution for cubic equations based on [Selby et al.-1962]. This close form solution for cubic equations was used to solve the unknown \( r \) in Eq. (A.5). No proof will be given.

A cubic equation, \( y^3 + py^2 + qy + r = 0 \), may be reduced to the form

\[
x^3 + ax + b = 0
\]

(A.32)

by substituting for \( y \) the value, \( x - \frac{p}{3} \), where

\[
a = \frac{1}{3}(3q - p^2) \quad \text{and} \quad b = \frac{1}{27} (2p^3 - 9pq + 27r).
\]
For solution, let

\[ A = \sqrt[3]{-\frac{b}{2} + \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}} \quad \text{and} \quad B = \sqrt[3]{-\frac{b}{2} - \sqrt{\frac{b^2}{4} + \frac{a^3}{27}}} \]

then the values of \( x \) will be given by

\[ x = A + B, \quad -\frac{A + B}{2} + \frac{A - B}{2}\sqrt{-3}, \quad -\frac{A + B}{2} - \frac{A - B}{2}\sqrt{-3}. \]

**Case 1** if \( \frac{b^2}{4} + \frac{a^3}{27} > 0 \), then there will be one real root and two conjugate imaginary roots.

**Case 2** if \( \frac{b^2}{4} + \frac{a^3}{27} = 0 \), then there will be three real roots of which at least two are equal.

**Case 3** if \( \frac{b^2}{4} + \frac{a^3}{27} < 0 \), then there will be three real and unequal roots.

In the last case, a trigonometric solution is useful. Compute the value of the angle \( \varphi \) in the expression

\[ \cos \varphi = -\frac{b}{2} \div \sqrt[3]{\frac{a^3}{27}}, \]

then \( x \) will have the following values:

\[ 2\sqrt{-\frac{a}{3}} \cos \left( \frac{\varphi}{3} \right), \quad 2\sqrt{-\frac{a}{3}} \cos \left( \frac{\varphi}{3} + 120^\circ \right), \quad 2\sqrt{-\frac{a}{3}} \cos \left( \frac{\varphi}{3} + 240^\circ \right). \]

### A.4 Close Form Solution for the Unknown \( r \) in Eq. (A.5)

Given the close form solution for cubic equations described in Section A.3, the unknown \( r \) in Eq. (A.5) can be solved directly instead of iteratively, as shown below.

From Eq. (A.4) and Eq. (A.5), i.e.,

\[ X_d + D_x = X_u \quad \text{and} \quad Y_d + D_y = Y_u \]
\[ D_x = X_d (\kappa_1 r^2), \quad D_y = Y_d (\kappa_1 r^2), \quad \text{and} \quad r = \sqrt{X_d^2 + Y_d^2} \]

we have

\[
\begin{align*}
  u^2 &= X_u^2 + Y_u^2 \\
  &= X_d^2 + 2D_x X_d + D_x^2 + Y_d^2 + 2D_y Y_d + D_y^2 \\
  &= (X_d^2 + Y_d^2) + 2\kappa_1 r^2 (X_d^2 + Y_d^2) + \kappa_1^2 r^4 (X_d^2 + Y_d^2) \\
  &= r^2 + 2\kappa_1 r^4 + \kappa_1^2 r^6 \\
  &= r^2 (1 + 2\kappa_1 r^2 + \kappa_1^2 r^4) \\
  &= r^2 (1 + \kappa_1 r^2)^2,
\end{align*}
\]

that is,

\[ u = \pm r \left( 1 + \kappa_1 r^2 \right). \]

However, because both \( u \) and \( r \) must be nonnegative, to ensure that \( u = r \) when \( \kappa_1 = 0 \) and \( r \neq 0 \), only one of the two cases holds; that is,

\[ u = r \left( 1 + \kappa_1 r^2 \right). \]

Notice that this implies that \( \kappa_1 \geq -1/r^2 \) for \( r > 0 \). Rearranging the terms, we have

\[ \kappa_1 r^3 + r - u = 0. \]

**Case 1** if \( r = 0 \), then \( u = 0 \).

**Case 2** if \( r > 0 \) and \( \kappa_1 = 0 \), then \( u = r \).

**Case 3** if \( r > 0 \) and \( \kappa_1 > 0 \), dividing both sides by \( \kappa_1 \), we get

\[ r^3 + \frac{1}{\kappa_1} r - \frac{1}{\kappa_1} u = 0. \]  \hspace{1cm} (A.33)
Noting that Eq. (A.33) is exactly in the same form as Eq. (A.32), the unknown \( r \) can be easily solved using the close form solution for cubic equations described in Section A.3.

### A.5 Computing Calibration Points

As we discussed earlier, the very first step in calibrating a camera is to prepare a calibration pattern, as illustrated in Figure A.6, grab a frame of the calibration pattern into the computer frame memory, and then compute the coordinate of each calibration point. We now present the algorithm to directly compute the coordinate of each calibration point \((X^i_j, Y^i_j)\)—or more precisely, the centroid of each circle—from the pattern image.

In what follows in this section, we assume that (a) The calibration pattern is a grid of black circles (or objects) on white background; (b) Each row has at least two circles, and there are at least two rows; and (3) The first two circles in the first row have a white hole inside. It is worth mention here that the algorithm, which we will shortly describe, not only works for circles, but it also works for other kinds of objects, provided that the size of each object is distinguishable from that of outliers, if any. Eventually, there are many ways to specify the calibration pattern. Circles are chosen here with an eye toward simplifying the computation of the centroid.

For simplicity of description, let us denote the pattern image by \( I_{pat} \), the low and high thresholds for the computation of the binary image and centroid by \( T_l \) and \( T_h \), the low and high thresholds for the removal of the outliers by \( A_l \) and \( A_h \). Let us also denote the intensity value at position \((x, y)\) in the pattern image by \( V \). Here now is the algorithm to compute the coordinate of each calibration point \((X^i_j, Y^i_j)\) in the pattern image.
1. Compute the binary image $I_{bin}$ from the pattern image $I_{pat}$ using the high threshold $T_h$. $T_l$ is ignored in the computation of the binary image.

2. Label the binary image using the Sequential Labeling Algorithm, as defined in [Horn-1986]. Call it $I_{lab}$.
3. Use the labeled image $I_{lab}$ as a reference to the pattern image $I_{pat}$ to compute both the area and centroid of all the circles in the pattern image $I_{pat}$, as follows:

$$A = \sum_{(x,y) \in R} w \cdot V, \quad X_f = \sum_{(x,y) \in R} w \cdot x / A, \quad \text{and} \quad Y_f = \sum_{(x,y) \in R} w \cdot y / A$$

where $R$ denotes the region of the circle with its centroid being computed, $A$ is the weighted area of the region $R$, $(X_f, Y_f)$ is the centroid of the region $R$, and $w$ is the weight of the pixel at position $(x, y)$ which is computed as follows:

$$w = \begin{cases} 
1, & \text{if } V < T_l, \\
0, & \text{if } V \geq T_h, \\
(T_h - V) / (T_h - T_l), & \text{otherwise.}
\end{cases}$$

The centroid thus computed are essentially the calibration points $(X_f, Y_f)$.

4. Remove outliers, if any, from the labeled image $I_{lab}$ using the thresholds $A_l$ and $A_h$. This is possible because we assume that the area of each circle in the pattern image is somewhere between $A_l$ and $A_h$.

5. Number the set of calibration points from left to right and top to bottom, with the circle having a hole inside and located at the upper left corner assigned the number 1. This step is necessary either to automatically match the coordinate of each calibration point to its corresponding world coordinate, or, in the case that when two or more images of the same calibration pattern are taken from different cameras, to match the same calibration point in different calibration images.

This algorithm is demonstrated in Figure A.7. Figure A.7a shows the pattern image. Figure A.7b shows the result of applying the algorithm described above to the image in Figure A.7a. The set of calibration points computed are marked by the + sign and
numbered as described in the last step of the algorithm. Moreover, our implementation of this algorithm generates not only the image coordinates of the calibration points but also the world coordinates corresponding to each calibration point. The resulting data can be fed directly to the camera calibration program described in [Chiang and Boult-1995].

**A.6 Distortion Correction**

We now introduce the algorithm for unwarping the images degraded by radial lens distortions. This algorithm is designed particularly for those applications that require sufficiently accurate warped intensity values to perform their tasks. It is imaging-consistent in the sense that will be described later in this thesis.

Given that the underlying camera model is determined, the algorithm is now for-
mulated, as follows:

**Step 1** Determine distorted sensor coordinates \((X_d, Y_d)\) using Eq. (A.6).

\[
X_d = d'_x(X_f - C_x)/s_x, \quad Y_d = d'_y(Y_f - C_y)
\]

where \(d'_x = d_x N_{cx}/N_{fx}\).

**Step 2** Determine undistorted sensor coordinates \((X_u, Y_u)\) using Eq. (A.4).

\[
X_u = X_d + D_x, \quad Y_u = Y_d + D_y
\]

where \(D_x = X_d(\kappa_1 r^2)\), \(D_y = Y_d(\kappa_1 r^2)\), \(r = \sqrt{X_d^2 + Y_d^2}\), and \((X_d, Y_d)\) is the distorted image coordinate on the image plane.

**Step 3** Determine undistorted image coordinates by substituting \((X_u, Y_u)\) determined in Step 2 for \((X_d, Y_d)\) in Eq. (A.6).

\[
X_f = s_xd'_x^{-1}X_u + C_x, \quad Y_f = d'_y^{-1}Y_u + C_y
\]

where \(d'_x = d_x N_{cx}/N_{fx}\).

**Step 4** Apply the imaging-consistent warping algorithms introduced in Chapter 3 to the undistorted image coordinates determined in Step 3 to unwarp radial lenses distortions.

This algorithm was used in Chapter 3 to show that pre-warping images to remove distortions can significantly improve the matching speed by allowing epi-polar constraints to result in a 1-D search. This idea is not new; however, previous work has presumed that bi-linear interpolation was sufficient for the warping. In Chapter 3, we showed how the integrating resampler can improve the quality of the matching results.
Figure A.8: Illustration of distortion correction. (a) original image; (b) (a) with radial lens distortions removed; (c) calibration pattern used to calibrate the camera and to correct the image shown in (a); (d) (c) with radial lens distortions removed.

An example is demonstrated in Figure A.8. Figure A.8a shows the original image; Figure A.8b, the result of applying the algorithm described above to the image in
Figure A.8a. To make it even easier to see the distortions and the results of applying the algorithm to the “distorted” images to get the “undistorted” images, Figure A.8c shows the calibration pattern used to calibrate the camera and to correct the image shown in Figure A.8a; Figure A.8d, the calibration pattern itself being corrected using the algorithm described above.

A.7 Conclusion

This chapter reviewed a two-stage algorithm for camera calibration using off-the-shelf TV cameras and lenses [Tsai-1986, Tsai-1987, Chiang and Boult-1995]. To make the work of measuring the image coordinates of the calibration points easier, we introduced an algorithm for automatically computing the image coordinates of the calibration points from the image of a calibration pattern. Then, for applications that require sufficiently accurate warped intensity values to perform their tasks, we presented an imaging-consistent algorithm for “unwarping” radial lens distortions once the underlying camera model is determined. Examples were given to show the use of these algorithms.