

THIN LENS-BASED GEOMETRIC SURFACE INVERSION FOR MULTIVIEW STEREO

A Dissertation
Presented to
The Academic Faculty

By

Robert Daniel Friedlander

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2020

Copyright © Robert Daniel Friedlander 2020

THIN LENS-BASED GEOMETRIC SURFACE INVERSION FOR MULTIVIEW STEREO

Approved by:

Dr. Anthony Yezzi, Advisor
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Patricio Vela
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Frank Dellaert
School of Interactive Computing
Georgia Institute of Technology

Dr. Justin Romberg
School of Electrical and Computer
Engineering
Georgia Institute of Technology

Dr. Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Date Approved: November 05, 2020

In loving memory of my grandparents, Harold and Ruth Stemerman.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank and acknowledge my advisor Dr. Anthony Yezzi; his insightful guidance, words of wisdom, and steadfast encouragement have allowed me to grow both professionally and personally and to reach this important milestone. I would also like to thank Drs. Patricio Vela and Frank Dellaert for their valuable comments and suggestions which have helped shape the course of my research, as well as Drs. Justin Romberg and Sung Ha Kang for reviewing this thesis. My colleagues in the Laboratory of Computational Computer Vision have also provided invaluable support and assistance, especially Huizong Yang who has been instrumental in helping me with image acquisition and camera calibration during these unprecedented times.

I am also grateful to the NDSEG Fellowship program, the National Science Foundation, the Army Research Office, and the Georgia Tech School of Electrical and Computer Engineering, all of which have helped fund my doctoral education and research.

Finally, I would like to express my gratitude to my brother and parents. It is because of their unending support, patience, encouragement, and love that I stand where I am today.

TABLE OF CONTENTS

Acknowledgments	iv
List of Tables	viii
List of Figures	ix
List of Symbols	xii
Chapter 1: Introduction and Background	1
1.1 Literature Survey	3
1.1.1 Shape-from-motion and multiview reconstruction	3
1.1.2 Photometric stereo and shape-from-shading	7
1.1.3 Shape-from-defocus	8
1.1.4 Deep learning approaches	9
1.1.5 Combining cues	9
1.2 Contributions	10
Chapter 2: Thin Lens Forward Model	12
2.1 Thin Lens Geometry	12
2.1.1 Geometry and Notation	12
2.2 Change of Variables Formulae	15

2.2.1	Method 1: Relating Corresponding Area and Solid Angle Elements	16
2.2.2	Method 2: Using Mixed Coordinates	18
2.3	Radiometry and the Image Irradiance Equation	22
2.3.1	Special Case: In-focus with Lambertian Reflectance (3D)	25
2.4	Comparison to Horn's Equation	28
2.4.1	Analytical Comparison	29
2.4.2	Experimental Comparison	33
Chapter 3: Thin Lens Model Inversion		40
3.1	Surface Evolution Equation	41
3.1.1	Radial/transverse coordinate frame	42
3.1.2	Irradiance sensitivity	42
3.1.3	Total matching error sensitivity	45
3.2	Scene Radiance Estimation	47
3.2.1	Smooth radiances	47
3.2.2	Constant radiances	48
Chapter 4: Experimental Results and Discussion		50
4.1	Qualitative Analysis	51
4.1.1	Synthetic Images	51
4.1.2	Real Images	55
4.2	Quantitative Analysis	57
Chapter 5: Conclusion and Future Work		59

Appendix A: Shape Sensitivity of Flux Through Shapes with Free Boundaries	62
A.1 Boundary integral	62
A.2 Surface integral	64
A.3 Combined result	65
Appendix B: Swapping Order of Integration for Occluding Boundary Integral	66
B.1 Occluding boundary and occluding projection	66
B.2 Dirac integration formula	67
B.3 Planar integration of occluding boundary to surface integration of occluding projection	67
Appendix C: Preliminary Computations (2D)	69
C.1 Full Jacobian for Change of Variables (2D)	69
C.1.1 Jacobian $\frac{\partial(s, \theta)}{\partial(x, \alpha)}$ of surface coordinates $s(x, \alpha)$ and $\theta(x, \alpha)$	69
C.1.2 Jacobian $\frac{\partial(x, \alpha)}{\partial(s, \theta)}$ of surface coordinates $x(s, \theta)$ and $\alpha(s, \theta)$	71
C.2 Image Irradiance Equation (2D)	73
C.3 Shape Sensitivity of Flux (2D)	74
C.3.1 Boundary terms	74
C.3.2 Contour Integral	75
C.3.3 Combined result	76
C.4 Surface Evolution Equation (2D)	76
References	88

LIST OF TABLES

2.1	Zemax Simulation Irradiance Errors (W/M^2)	35
2.2	Analysis of Model Fitting Parameters	37
2.3	Analysis of Accuracy Differences in Modeled Irradiance	38
2.4	Shape-from-Shading Simulation Results	39
4.1	Performance Comparison of Thin Lens and Pinhole Reconstruction Methods	57
C.1	2D Angle Relationships	69

LIST OF FIGURES

2.1	Geometry of the thin lens imaging model. A cone of rays from each light-emitting point is captured by the lens and focused to a point behind the lens whose depth is related to the depth of the emitter. If the image plane is incorrectly positioned, the image of the emitter is blurred out from the divergence of the light rays at the image plane.	13
2.2	All rays through lens leaving P are focused at P' . The solid rays are the three principal rays from the thin lens axioms. Adapted with permission from [100] © 2017 IEEE.	14
2.3	Any ray can be parameterized using either surface coordinates or focal plane coordinates. The shaded region represents the tangent plane to the surface at $S(s_1, s_2)$, and the dashed line represents the optical axis. While the ray shown here goes through the center of the lens O , there is a whole cone of rays that pass through both $P(X)$ and the lens as denoted by the dotted lines, and the size of this cone is determined by the focus depth z and lens diameter d . Adapted with permission from [100] © 2017 IEEE. . .	15
2.4	Solid angle subtended by a patches on the focal plane and surface as seen from the lens center. Adapted with permission from [101] © 2020 The Optical Society.	17
2.5	Solid angle subtended by a lens patch as seen from both the surface and the focal plane. Adapted with permission from [101] © 2020 The Optical Society.	18
2.6	Geometry used by Horn to relate surface radiance to image irradiance. Only rays through the lens center are considered. Adapted with permission from [100] © 2017 IEEE	29
2.7	The Horn and lens irradiance equations are near identical for small apertures and far-focused points but diverge for near-focused and centered points as the aperture increases. This becomes significant for f -numbers less than about 2. Reused with permission from [101] ©2020 The Optical Society . .	31

2.8	Cropped photo taken of LED point source focused onto one pixel. Reused with permission from [101] ©2020 The Optical Society.	36
4.1	Focused (left) and defocused (right) images from the first tetrahedron dataset (2 of 30 views). Defocus blur makes sharp corners appear rounded and causes the tetrahedron to appear larger than its actual size	50
4.2	Two viewpoints of the thin lens-based reconstruction (top) and the pinhole-based reconstruction (bottom) for the first tetrahedron dataset. The sharp edges and corners of the shapes were reconstructed accurately using the thin lens method but rounded using the pinhole method	51
4.3	Four snapshots of the evolving surface for first tetrahedron dataset using the thin lens method after 0, 20, 60, and 120 iterations	51
4.4	Two viewpoints of the thin lens-based reconstruction (top) and the pinhole-based reconstruction (bottom) for the second tetrahedron dataset. The thin lens method successfully reconstructed the single tetrahedron while the pinhole method could not reconcile the two different thicknesses visible in the images	52
4.5	Focused (left), defocused (middle), and noisy and defocused (right) images from the dumbbell dataset (2 of 30 views). Defocus blur makes the entire object appear wider and the disk edges appear rounded instead of being a sharp transition	53
4.6	Two viewpoints (left and middle) of the thin lens-based reconstruction (top) and the pinhole-based reconstruction (bottom) for the dumbbell dataset. The thin lens method more accurately reconstructs the sharp edges of the disks and the actual thickness of both the disks and the central rod when compared with the pinhole method. When noise was added to the dumbbell dataset (right), the thin lens reconstruction was barely affected while the pinhole reconstruction was of noticeably lower quality	54
4.7	Four snapshots of the evolving surface for the dumbbell dataset using the thin lens method after 0, 160, 240, and 400 iterations	54
4.8	Zoomed-in focused (left) and defocused (right) images from the chesspiece dataset (2 of 32 views). Sharp features like occluding edges and texture on the mane are blurred out and indistinguishable in the defocused images. . .	55

4.9	Two viewpoints of the thin lens-based (top) and pinhole-based reconstructions (bottom) for the knight dataset. The thin lens method correctly reconstructs the shape of the knight and is able to better carve out the slot between the muzzle and the body compared to the pinhole method.	56
4.10	Four snapshots of the evolving surface for the chesspiece dataset using the thin lens method after 0, 20, 40, and 70 iterations	56

LIST OF SYMBOLS

O	Center of lens (and spatial origin)
f	Focal length of lens
d	Diameter of lens
Ω'	Image domain
X'	2D image plane coordinate = (x'_1, x'_2)
z'	Image plane depth (negative)
P'	Point (X', z') in image plane
dX'	Image plane area element
Ω	Conjugate image domain
X	2D focal plane coordinate = (x_1, x_2)
z	Focal plane depth (positive)
P	Point (X, z) in focal plane
dX	Focal plane area element
e_z	Unit normal to focal plane
s_1, s_2	Isothermal surface coordinates
u, v	Time-independent surface coordinates
S	Surface point
dS	Surface area element
S^*	Occluding boundary point
ds^*	Occluding boundary length element
N	Outward unit normal to surface

G	Background point
η, γ	Angular background coordinates
Γ	Set of all detected light rays
Γ_S/Γ_G	Set of detected rays emitted from surface/background
θ	Ray angle with respect to N
ϕ	Ray rotation angle around N -axis
$d\theta$	Solid angle element w.r.t. surface = $\sin \theta d\theta d\phi$
α	Ray angle with respect to e_z
β	Ray rotation angle around e_z
$d\alpha$	Solid angle element w.r.t. focal plane = $\sin \alpha d\alpha d\beta$
r	Ray length = $\ S - P\ $
e_r	Unit ray vector = $(S - P)/r$
L	Surface radiance
K	Background radiance
$d\mathcal{P}$	Radiant power element
\mathcal{P}	Total detected radiant power
E	Conjugate irradiance (through focal plane)
dQ	Conjugate irradiant power element
Q	Total conjugate irradiant power
E'	Image irradiance (through image plane)
dQ'	Image irradiant power element
Q'	Total image irradiant power
I	Measured image
\mathcal{E}	Conjugate error = $M^{-2}E(X) - I(MX)$
\mathcal{E}'	Image error = $E'(X') - I(X')$
C	Number of images
J	Total squared-error = $\sum_{c=1}^C \frac{1}{2} \int_{\Omega'_c} \mathcal{E}'(X'_c)^2 dX'_c$

SUMMARY

In this thesis, a fully generative algorithm is developed for the reconstruction of dense three-dimensional shapes from scene images under varying viewpoints and levels of focus. Current state-of-the-art multiview methods are founded on a pinhole camera model that assumes perfectly focused images and thus fail when given defocused image data. The method developed herein overcomes this by instead assuming a thin lens which is able to accurately model defocus blur in images. While easily stated, this requires a significant mathematical reformulation from the bottom up as the simple perspective projection assumed by the pinhole model and utilized by current methods no longer applies under the more general thin lens model. New expressions for the forward modeling of image formation as well as model inversion are developed. For the former, image irradiance is related to scene radiance using energy conservation, and the resulting integral expression has a closed-form solution for in-focus points that is shown to be more general and accurate than the one used in current methods. For the latter, the sensitivities of image irradiance to perturbations in both the scene radiance and geometry are analyzed, and the necessary gradient descent evolution equations are extracted from these sensitivities. A variational surface evolution algorithm is then formed where image estimates generated by the thin lens forward model are compared to the actual measured images, and the resulting pixel-wise error is then fed into the evolution equations to update the surface shape and scene radiance estimates. This algorithm is experimentally validated for the case of piecewise-constant scene radiance on both computer-generated and real images, and it is seen that this new method is able to accurately reconstruct sharp object features from even severely defocused images and has an increased robustness to noise compared to pinhole-based methods.

CHAPTER 1

INTRODUCTION AND BACKGROUND

The problem of acquiring knowledge about the spatial geometry of objects of interest from direct scene measurements, such as two-dimensional (2D) images, has been a central pillar in computer vision research for decades; the field of 3D reconstruction deals with developing algorithms to solve this problem. Applications for 3D reconstruction are varied and many. In medicine, it can be used as an aid for diagnosis or to construct patient-specific molds, while in robotics it can be used to help a machine understand its surroundings. It can also be used to generate digital models when CAD software is unavailable or as an initial starting point for CAD designers to then modify to desired specifications. In artifact preservation, 3D reconstruction can be used as a non-destructive method to create a model of an fragile or valuable artifact; this model can be 3D printed or otherwise manufactured and displayed in place of the actual artifact. Of particular importance in recent years is the application to virtual and augmented reality, where realistic computer models of objects and environments is needed.

Since the scene measurements used in 3D reconstruction are often optical images, reconstruction methods need to specify an assumed model of image formation. The model used in most methods (with the exception of shape-from-defocus, described below) is the pinhole camera model, where the camera lens is modeled as an infinitesimally small hole. This allows a point in space to be mapped to its corresponding image point via a projection operator which can be easily represented using linear algebra. Additionally, any point in space visible from the pinhole projects to a unique point in the image. Physically, this means even though light may be emitted from a point in multiple directions, only one ray in one direction will actually pass through the pinhole and reach the image plane. This property of the pinhole model implies that the entirety of any image formed with this model is

perfectly in-focus. In practice, however, many lenses are only well-focused within a small depth-of-field, and algorithms based on the pinhole model lose accuracy and applicability in such situations. For instance, if some images exhibit image blur in textured regions due to defocus, these algorithms will treat this blur as a property of the scene, causing these reconstructed regions to be smoother and have less texture than they do in reality. There are also applications, such as endoscopy, where it is not practical to assume either the camera can be held in a steady position long enough or the scene is static enough to acquire focused images. A more general image formation model that takes refraction and limited-focus into account, such as the thin lens model, treats blur naturally as a property of the imaging geometry, not the scene. Thus, a reconstruction method that uses a thin lens model should work well for images exhibiting defocus blur.

Such a method helps to unify the many approaches taken to 3D reconstruction over the years. These approaches can be categorized by the imaging parameter that varies between each image, known as the cue. Cues can be any number of things, but the three most often used are camera viewpoint, scene illumination, and focus depth. For instance, in *multiview reconstruction* (also known as *stereo reconstruction*) the images are captured from cameras placed at different positions but with identical intrinsic settings, while in *photometric stereo* (also known as *shape-from-shading* in the single image case) and *shape-from-defocus* it is the lighting conditions and focus settings of the camera that change for each image, respectively, while the camera stays in a fixed position.

Despite this artificial segmentation of 3D reconstruction into one-cue methods, there is no reason that reconstruction cannot be done while considering multiple cues. This work aims to address the problem of reconstructing dense objects from scene measurements under varying viewpoints and levels of focus by developing a new class of variational algorithms utilizing the thin lens optical model. This is done by retaining the general methodology of existing surface inversion methods for multiview reconstruction but replacing the currently used pinhole model with the thin lens model, a change that is not trivial and re-

quires substantial mathematical reformulation. The jump in model complexity allows for better reconstruction when the available images are not well-focused. It also provides a unified framework where different image cues (e.g. viewpoint or focus) are treated equally, and the one-cue “shape-from-X” problems fall out as special cases.

1.1 Literature Survey

An overview of the primary frameworks for and approaches to 3D reconstruction taken over the last 50 years is presented below, with an emphasis on multiview reconstruction as this is the most aligned with the work presented here. Each general approach is explained and key algorithmic advances noted. Additional information regarding the general classes of methods, different surface representations used, and specific state-of-the-art algorithms can be found in [1, 2, 3, 4].

1.1.1 Shape-from-motion and multiview reconstruction

The earliest proposed techniques for 3D reconstruction dealt with *shape-from-motion*, starting with the methods proposed by Adiv [5] and Faugeras [6]. Shape-from-motion works by tracking the motion of brightness patches through a succession of images [7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Such motion can either be caused by actual movement in the scene or by the movement of the camera, the latter of which leads to the case of multiview reconstruction. One of the earliest and more popular of these methods, still taught to this day, utilizes the simple epipolar geometry provided by the pinhole camera model as well as assumptions of Lambertian reflectance [17, 18]. It uses a basic triangulation strategy by finding sets of corresponding points in the input images and then backtracing along the rays until the point of intersection is found. This results in a final object representation of a point cloud or cluster of spatial points where some additional processing needs to be carried out to interpolate between the points to form a final connected surface. In the ideal case, this intersection exists and is unique; however due to numerics and image noise the rays in

practice may not even intersect. Methods that rely on epipolar geometry like this typically differ most in how they find corresponding image points [19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]. The quality of this point-matching is the main factor in the accuracy of these methods. Thus they work best for well-focused scenes and objects with high levels of texture where the point-matching is the least ambiguous. When the images have many homogeneous or repetitive regions, the point-matching problem becomes ill-posed. Probabilistic methods can be employed in these cases instead of the usual feature-detection techniques; for instance, Markov Chain Monte Carlo methods are used in [34, 35, 36, 37] to find the optimal set of point correspondences.

Epipolar matching can be taken even further by allowing for iterative refinement of the reconstructed points. The collection of “bundle adjustment” methods uses the set of matched image points to simultaneously reconstruct the 3D coordinates of the corresponding scene points as well as the camera calibration parameters; this is done through an gradient descent process aiming to minimize the reprojection error [38]. Among other things, this helps to counter poor or unknown camera calibrations as well as noise. Even with this however, algorithms depending on epipolar matching ignore much of the available image information since the number of matched points is typically a small fraction of the total number of pixels in each image.

Some methods avoid epipolar matching altogether and instead directly use the image data in an attempt to represent the imaged environment in a hierarchical fashion. The conceptually simplest of these methods represents the scene as a set of parallel planes at different depths and decomposes the images into “layers” of pixels, each layer on a different depth plane [39], essentially forming a depth map for each image. Another set of methods take a “plane+parallax” approach by warping and compensating for a reference plane. The residual planar parallax, or the deviations from this reference frame, can be used to compute depth values for image points not on the plane [40, 41]. This is taken further in [42] where a parametric surface (not necessarily planar) is used as the reference.

It was shown in [43] that parallax representations are helpful in calibrating cameras and in [44] that simultaneous camera calibration and 3D reconstruction can be done using parallax approaches. These parallax representations allow for large image mosaics and panoramas to be computed as the reference plane allows easy image alignment and texture-mapping. Such scene representations are also useful in the field of image-based rendering.

While the methods discussed so far use image data to construct a representation of the scene, methods based on voxel coloring and space carving can be thought of as going in the opposite direction. The scene is first represented by a volume of voxels which are then traversed one at a time, coloring and/or carving away the voxels as to stay photoconsistent with the input images. In this context, photoconsistent means both that the shape represented by the voxels produces images close to the input images and that each individual voxel results in a similar pixel value for all cameras to which it is visible. The end result is a volumetric collection of voxels that contains all possible photoconsistent shapes, called the “photo hull” [45]. Voxel coloring is primarily concerned with the coloring of voxels while space carving simultaneously carves and colors.

If prior information is known about the objects being reconstructed, this can be incorporated into the reconstruction algorithm to provide a higher level of accuracy and robustness. Examples of this can be seen in areas such as architecture modeling [46] and 3D medical imaging [47], where the class of objects considered share common basic shapes or characteristics.

Variational stereo

The problem of multiview reconstruction has also been viewed under a variational framework. Instead of waiting to reconstruct the 3D scene geometry until after sufficient image analysis has occurred (as in the epipolar matching methods), these variational methods start with an initial set of piecewise smooth surfaces that are iteratively deformed. The deformation is governed by a gradient descent process aiming to minimize an energy functional

representing some error measure between the surface estimates and the given data. This is similar in concept to space carving, except for one key difference. In space carving, once a voxel has been eliminated from the surface model, it is never reconsidered; in these variational methods, however, the surface is allowed to evolve in whatever direction yields the lowest energy, meaning a voxel eliminated in one iteration could be added back to the surface in some subsequent iteration. This energy minimization is done by writing the gradient flow as the solution to a geometric partial differential equation (PDE), where an artificial time parameter represents iteration time. Such PDEs can be implemented easily using the level set numerical methods developed by Osher and Sethian in [48, 49], where the surface is embedded as the zero level set of a higher-dimensional function. These level set methods handle topological changes in the evolving manifold without any special intervention from the user, and commonly used geometric quantities such as unit normals and curvature are easily computable.

The first variational methods were proposed by Faugeras and Kerivan in [50] and further elaborated upon in [51]. In the Lambertian and noiseless case, images of the same 3D scene point taken from different cameras in different positions should exhibit the same image value at the corresponding image points (which can be found via a perspective projection of the scene point onto the camera planes); this is simply an extreme version of photometric consistency. Thus, an intuitive error measure for a single surface point would be to consider the squared-error between the values of each pair of projected image points; this can then be integrated over the entire surface for an overall error measure that is used as the energy functional. In order to counter sensitivity to noise and outliers (such as small areas of dense texture), it was proposed to adjust this functional into a cross-correlation measure instead of a direct error measure [50]. One can consider this as framing epipolar matching as a variational problem as it still uses direct comparisons between image points as a basis for the energy functional though it utilizes significantly more of the information available in the images.

This initial variational work was taken further by Yezzi and Soatto in [52, 53]; they developed a fully generative variational method for multiview reconstruction by framing the problem as one of 3D segmentation. Along with estimating the scene geometry, this method also estimates the scene radiance. Assuming the cameras are calibrated, this allows for synthetic images to be generated from the scene estimate through a forward model that can be compared to the input images, providing an error measure that directly compares the estimated quantities with the data. The resulting (regularized) energy functional can then be written as an integral over the image domains rather than over the surfaces, which provides simpler computation, greater numerical stability, and greater robustness to specular reflections as shown in [52, 53, 54, 55, 56, 57]. This approach also filled a complementary role to epipolar matching methods since it performs well for scenes with sparse or repeating textures, which as discussed above cause the point correspondence problem to be ill-posed [52]. While the original method was developed under strict assumptions (Lambertian reflectance, constant albedo, calibrated cameras, consistent lighting/exposure settings, and perfect focus), it has been expanded upon over the last two decades to allow these assumptions to be gradually relaxed without changing the underlying methodology [58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71], except for the focus assumption which is the subject of the research herein. It has also been modified for some application-specific settings, such as ocean wave modeling [54, 55, 56, 57] and cardiac CT segmentation [47]. It has also been extended recently to work with radar signals [72]. Additionally, another variational approach has been taken by Oswald in [73] where differences in image acquisition time, as well as viewpoints, are accounted for.

1.1.2 Photometric stereo and shape-from-shading

As alluded to above, photometric stereo encompasses the collection of methods that infer surface orientation from multiple images of an object from a single viewpoint but under varying lighting conditions. It was shown by Horn in [74] that this problem in the sin-

gle image case can be framed as a nonlinear first-order partial differential equation (PDE) dependent on the image data, the known illumination, and a model of the bi-directional reflectance function (BRDF); this PDE is called the brightness or irradiance equation (or the rendering equation, in the realm of computer graphics). This started the lengthy work in the field of shape-from-shading, and from this, Woodham proposed the method of photometric stereo for multiple images [75]. As such, algorithms for photometric stereo and shape-from-shading set themselves apart by how they approach solving this irradiance PDE [74, 76, 77, 78, 79, 80, 81, 82, 83]. While initial approaches did assume Lambertian reflectance, state of the art methods for photometric stereo have found ways to account for non-Lambertian surfaces [84].

1.1.3 Shape-from-defocus

Methods for shape-from-defocus (and its cousin shape-from-focus) take a unique approach compared to the other forms of 3D reconstruction already discussed; namely, the internal properties of the camera are varied while all external camera and scene properties are assumed constant. Here the set of input images are taken by a camera in a single position, but the focus setting of the camera is changed for each image. Thus, the well-focused portion of each image (e.g. visually crisp and not blurry) represents the portion of that scene that exists at a specific depth that is easily computed from knowledge of the camera's focal properties. In this manner, each image contains information about a specific depth slice, and the full surface can be reconstructed by drawing out and aggregating this depth information. The pinhole camera model cannot be used here as it cannot model the limited depth of field of real lenses. Instead, shape-from-defocus methods typically assume either a thin lens model [85, 86] or a Gaussian or other estimated optical kernel [87, 88], all of which can model defocus blur. Additionally, some efforts have been made to solve the shape-from-defocus problem in a variational framework [89, 90], and the work developed in this dissertation can be seen as a generalization of these methods that relaxes the

fixed-viewpoint assumption and thus gain additional spatial information from parallax.

1.1.4 Deep learning approaches

The rise of ever more powerful processors and the decreasing cost of GPUs has allowed some attempts at applying deep learning to 3D reconstruction. For instance, deep convolutional-neural networks are trained to carry out depth-from-defocus on single input images in [91, 92] as well as multiview reconstruction [93, 94, 95, 96]. A thorough overview of the deep learning methods developed over the last half-decade can be found in [4] and the references therein. While the training of deep learning systems can be time-consuming and computationally intensive, sometimes taking days or even weeks, such systems have the benefit of generalizability and fast run-time speeds. This is in contrast to more traditional approaches to 3D reconstruction which in essence have to start from scratch for every new application. Deep learning can also form the first part of a cascaded reconstruction system where the output of the network is passed as the initial surface estimate to a variational method, such as the method developed here.

1.1.5 Combining cues

A small number of attempts have been previously made to integrate both stereo and defocus cues. In [97], depth-from-focus is used to generate disparity maps for each viewpoint which are then combined using stereo correspondence. Because of this, it has all the limitations of other point correspondence methods, which our algorithm avoids. Also, our method consists of a single process that inherently and naturally integrates both cues, unlike the cascaded single-cue structure in [97]. Similarly, in [98], stereo disparity maps are computed using point correspondence between images taken with a short baseline. If defocus is present, the disparity maps are refined using estimated Gaussian blur kernels. However, this method requires a specialized 3-camera system for each viewpoint, unlike our method which only needs a single camera, and the final output is a sequence of disparity maps,

not a full dense 3D model. Finally, in [99] a method combining depth-from-defocus and multiview reconstruction is proposed for light-field cameras. While effective, this method requires the specialized sensor microarrays found in light-field cameras and cannot be used with conventional CCD cameras which are more prevalent and widely accessible.

1.2 Contributions

Chapter 2 deals with the development and verification of the thin lens forward model, which expresses image irradiance as a function of scene radiance and geometry. While the equation governing this model is in general an integral, it has a closed-form solution for the special case of an in-focus point with Lambertian reflectance. This novel irradiance-radiance relationship resulted in a conference presentation [100] and journal publication [101], where it was shown to be more general and accurate than the relationship first proposed by Horn [102] that is still used today in state-of-the-art methods. The contents and results of these papers are addressed along with some additional developments which are used later on in this thesis.

Chapter 3 deals with the development of the thin lens model inversion, which is the crux of this novel reconstruction algorithm that allows for reconstruction from defocused images from multiple viewpoints, even if the focal parameters change between images. The energy functional is formed as the total squared-matching error between the input images and the synthetic images (which are generated using the forward model from Chapter 2). The energy's gradient flow is then derived by looking at the sensitivity with respect to the surface shape and scene radiance, the former of which is not a trivial computation. It will be shown that in the case of piecewise-constant scene radiance the gradient flow is non-zero only for surface points that contribute to occluding boundary projections in the images.

Chapter 4 contains the results of applying the thin lens-based reconstruction to both synthetic images generated using Blender as well as real images. It will be seen that this novel thin lens-based method is able to recover sharp features, that are blurred and rounded

in the images due to defocus, significantly better than pinhole-based methods. Also, the thin lens-based method is able to accurately recover object shape even when the level of defocus is varied among the images, a scenario where pinhole-based methods fail due to being unable to reconcile the resulting varying object sizes in the images. While using the thin lens model increases the algorithm's computational complexity compared to pinhole-based methods, it does cause a marked decrease in convergence time with respect to the number of iterations required. At the time of this writing, manuscripts are being prepared using the contents of Chapters 3 and 4 for eventual conference and journal submission.

CHAPTER 2

THIN LENS FORWARD MODEL

In this chapter, the thin lens optical model will be explained and the irradiance equation governing the corresponding forward model will be developed. This equation is a key component in the fully-generative variational reconstruction framework, allowing synthetic images based off the current shape estimate to be generated and compared to the actual image data. The irradiance equation developed here is a more general version of the classical equation developed by Horn, and both analytical and experimental comparisons of these two equations are presented at the end of this chapter.

2.1 Thin Lens Geometry

2.1.1 Geometry and Notation

Figure 2.1 gives a visual depiction of the thin lens imaging model. We consider a lens of diameter d and focal length f , where the center of the lens O acts as the origin for our coordinate system. A thin lens is capable of capturing light over a finite solid angle of directions and focusing that light to a specific point behind itself (through refraction).

The three primary axioms of the thin lens model determine the location of this point. First, light rays entering the lens parallel to the optical axis converge behind the lens on the optical axis at the focal point, which lies at depth $-f$. Second, rays emitted from the focal point that pass through the lens exit the lens parallel to the optical axis. Third, rays through the center of the lens are not refracted. When put together, these axioms are the foundation for the thin lens equation, which states that a thin lens with focal length f perfectly focuses all points at depth z (sometimes known as the *focus depth*) onto an image plane at depth $-z'$ such that:

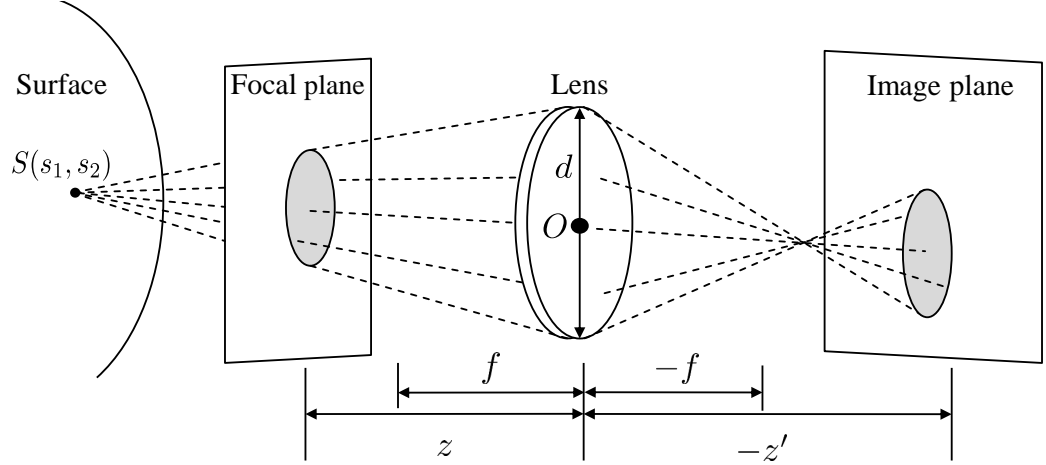


Figure 2.1: Geometry of the thin lens imaging model. A cone of rays from each light-emitting point is captured by the lens and focused to a point behind the lens whose depth is related to the depth of the emitter. If the image plane is incorrectly positioned, the image of the emitter is blurred out from the divergence of the light rays at the image plane.

$$\frac{1}{f} = \frac{1}{z} - \frac{1}{z'} \quad (2.1)$$

This plane of focus at depth z is called the *focal plane*. Since points on the focal plane are perfectly focused, all light rays emitted from a point $P(X) = (X, z)$ on the focal plane converge at a point $P'(X') = (X', z')$ on the image plane, as shown in Figure 2.2, where the solid rays demonstrate the three thin lens axioms. It is important to note that the third axiom guarantees that the line going through both P and P' must also go through the center of the lens. Furthermore, the 2D planar coordinates $X = (x_1, x_2)$ and $X' = (x'_1, x'_2)$ on the focal and image planes, respectively, and their corresponding area elements $dX = dx_1 dx_2$ and $dX' = dx'_1 dx'_2$ are related as follows:

$$X' = MX \quad (2.2a)$$

$$dX' = M^2 dX \quad (2.2b)$$

where $M = z'/z = 1/(1 - z/f)$ is the lens magnification.

For general points, in particular those not on the focal plane, a little more notation must

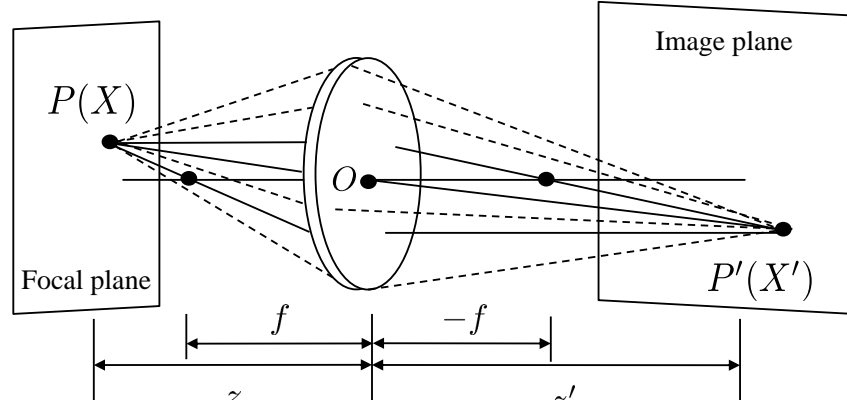


Figure 2.2: All rays through lens leaving P are focused at P' . The solid rays are the three principal rays from the thin lens axioms. Adapted with permission from [100] © 2017 IEEE.

be introduced. Consider a surface point $S(s_1, s_2)$ not on the focal plane, where (s_1, s_2) are isothermal coordinates for the surface. The shape of the image of this point in the image plane can be found easily using a conjugate image in the focal plane. First, all the rays emitted from $S(s_1, s_2)$ that pass through the lens are traced to find where they intersect the focal plane, forming the conjugate image; then the conjugate image is mapped to the image plane using the coordinate relationship (2.2a). This is shown in Figure 2.1 where a single unfocused point on the surface forms a circle in the image instead of a point.

The set of all emitted rays that pass through the lens is denoted as Γ and can be parameterized in two convenient ways as illustrated in Figure 2.3, where the shaded region is the tangent plane to the surface at $S(s_1, s_2)$. Using surface coordinates (s_1, s_2, θ, ϕ) , a ray of length r in direction $e_r = (S - P)/r$ can be parameterized by the surface point $S(s_1, s_2)$ from which it was emitted, the elevation angle θ the ray makes with the surface normal $N(s_1, s_2)$, and the azimuth angle ϕ in the tangent plane (e.g. the rotation around N). Likewise using focal plane coordinates $(x_1, x_2, \alpha, \beta)$, a ray can be parameterized by the point $P(x_1, x_2)$ at which it intersects the focal plane, the angle α between the ray and the focal plane unit normal e_z , which is parallel to the optical axis, and the azimuth angle β in the focal plane (e.g. the rotation around e_z). Equivalently, the angular component of

these parameterizations can be written in terms of the solid angles subtended by the set of rays as seen from the surface and the focal plane, denoted θ and α respectively. Without loss of generality it can be assumed for the above that the focal plane lies in front of the surface point. If it instead lies behind the surface, the conjugate image is virtual and rays have to be virtually backtraced to intersect the focal plane.

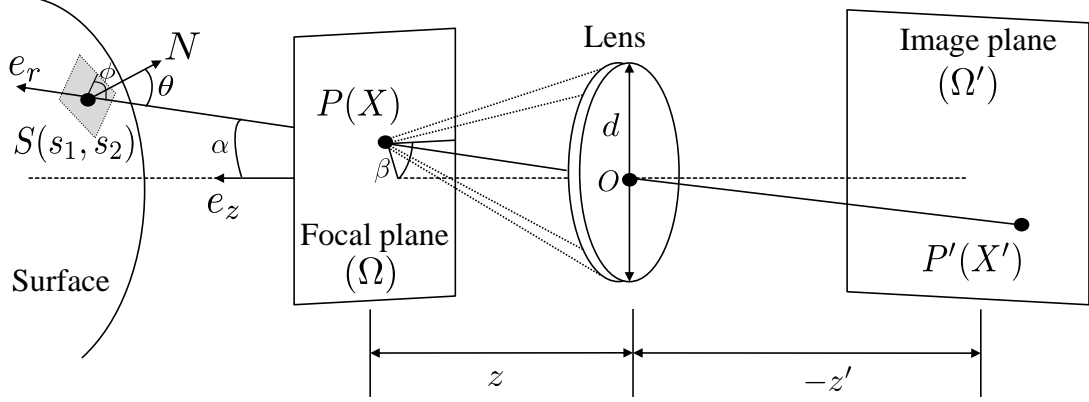


Figure 2.3: Any ray can be parameterized using either surface coordinates or focal plane coordinates. The shaded region represents the tangent plane to the surface at $S(s_1, s_2)$, and the dashed line represents the optical axis. While the ray shown here goes through the center of the lens O , there is a whole cone of rays that pass through both $P(X)$ and the lens as denoted by the dotted lines, and the size of this cone is determined by the focus depth z and lens diameter d . Adapted with permission from [100] © 2017 IEEE.

2.2 Change of Variables Formulae

It will be particularly convenient, both here as well as in Chapter 3, to rewrite integrals in surface coordinates $(S, \theta) = (s_1, s_2, \theta, \phi)$ as integrals in focal plane coordinates $(X, \alpha) = (x_1, x_2, \alpha, \beta)$, and vice versa. The full Jacobian matrix between these two sets of coordinates for the simpler 2D case is given in Appendix C.1. Computing the full Jacobian matrix for the 3D case would be unwieldy and involve many unnecessary terms that would cancel out in the end as seen in the 2D case. Despite this, it is possible to compute the change of measure between the surface and focal plane coordinates without going through the full Jacobian matrix. Two such computations are shown here. First, the method presented in

[101] will be given, where the area elements and solid angle elements of the two coordinate systems are related separately and then multiplied to find the final change of measure. Second, mixed coordinates $(S, X) = (s_1, s_2, x_1, x_2)$ will be used as an intermediate coordinate system, and the Jacobians between these mixed coordinates and the surface and focal plane coordinates will be found. Both of these methods allow the final change of measure to be computed by only considering one type of variable (spatial or angular) at a time.

2.2.1 Method 1: Relating Corresponding Area and Solid Angle Elements

Consider the situation presented in Figure 2.4. Here a small patch on the surface, centered at $S(s_1, s_2)$ (at depth \tilde{z}) and of area dS , is illuminated by a cone extending from the lens center. This same cone also illuminates a small patch on the focal plane, centered at $P(X)$ and of area dX . This cone is at an angle θ relative to the surface normal N and α relative to the focal plane normal e_z . The solid angles subtended by both the surface and focal plane patches from the point of view of the lens center are given by

$$d\omega_{\text{lens,surf}} = \frac{dS \cos \theta}{(\tilde{z} / \cos \alpha)^2} = \frac{dS \cos \theta \cos^2 \alpha}{\tilde{z}^2} \quad (2.3a)$$

$$d\omega_{\text{lens,fp}} = \frac{dX \cos \alpha}{(z / \cos \alpha)^2} = \frac{dX \cos^3 \alpha}{z^2} \quad (2.3b)$$

The solid angle subtended by both patches, from the point of view of the lens center, are equal, so we can equate (2.3a) and (2.3b) to yield the following relationship between dS and dX :

$$dX = \left(\frac{z}{\tilde{z}}\right)^2 \left(\frac{\cos \theta}{\cos \alpha}\right) dS \quad (2.4)$$

Now consider the situation presented in Figure 2.5. Here a small patch on the lens of area dA_L is illuminated by a cone of rays emitted from a surface point $S(s_1, s_2)$. As seen from the point of view of the surface at $S(s_1, s_2)$, this lens patch subtends a solid angle $d\theta$ given by

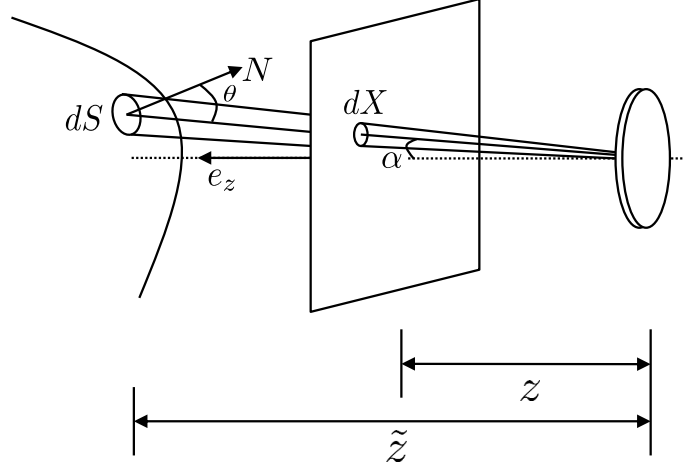


Figure 2.4: Solid angle subtended by a patches on the focal plane and surface as seen from the lens center. Adapted with permission from [101] © 2020 The Optical Society.

$$d\theta = \frac{dA_L \cos \alpha}{(\tilde{z}/\cos \alpha)^2} = \frac{dA_L \cos^3 \alpha}{\tilde{z}^2} \quad (2.5)$$

where \tilde{z} is again the depth of $S(s_1, s_2)$ along the optical axis. However, the solid angle that the lens patch subtends can also be viewed from the perspective of the point $P(X)$ on the focal plane, and this solid angle $d\alpha$ is given by

$$d\alpha = \frac{dA_L \cos \alpha}{(z/\cos \alpha)^2} = \frac{dA_L \cos^3 \alpha}{z^2} \quad (2.6)$$

Since the area of the lens patch stays constant, (2.5) and (2.6) can both be solved for dA_L and then equated to yield

$$d\alpha = \left(\frac{\tilde{z}}{z}\right)^2 d\theta \quad (2.7)$$

Finally, multiplying (2.4) with (2.7) gives the change of measure

$$dX d\alpha = \left(\frac{z}{\tilde{z}}\right)^2 \left(\frac{\cos \theta}{\cos \alpha}\right) dS \left(\frac{\tilde{z}}{z}\right)^2 d\theta = \left(\frac{\cos \theta}{\cos \alpha}\right) dS d\theta \quad (2.8)$$

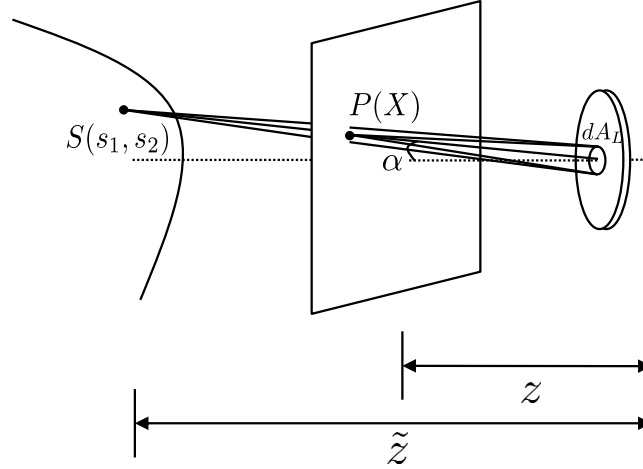


Figure 2.5: Solid angle subtended by a lens patch as seen from both the surface and the focal plane. Adapted with permission from [101] © 2020 The Optical Society.

2.2.2 Method 2: Using Mixed Coordinates

Focal plane coordinates $(x_1, x_2, \alpha, \beta)$ to mixed coordinates (x_1, x_2, s_1, s_2)

For a fixed point $P(x_1, x_2)$ in the focal plane, if the elevation angle α of the ray unit vector e_r with respect to the focal plane unit normal e_z is varied, we obtain a local parameterization of the visible portion of the surface S in terms of α and β . Using a spherical coordinate system with origin P , e_r can be written solely in terms of α and β . This allows the following orthonormal basis for \mathbb{R}^3 to be obtained (which can also be written solely in terms of α and β):

$$\begin{aligned} e_r &= (\sin \alpha \cos \beta, \sin \alpha \sin \beta, \cos \alpha) \\ \frac{\partial e_r}{\partial \alpha} &= (\cos \alpha \cos \beta, \cos \alpha \sin \beta, -\cos \alpha) \\ e_r \times \frac{\partial e_r}{\partial \alpha} &= (-\sin \beta, \cos \beta, 0) \end{aligned}$$

If we differentiate e_r with respect to β , we get a similar orthogonal basis, where each of the original basis vectors is now multiplied by either a positive or negative factor of $\sin \alpha$:

$$\begin{aligned}
\frac{\partial e_r}{\partial \beta} &= \sin \alpha (-\sin \beta, \cos \beta, 0) = \left(e_r \times \frac{\partial e_r}{\partial \alpha} \right) \sin \alpha \\
e_r \times \frac{\partial e_r}{\partial \beta} &= -\sin \alpha (\cos \alpha \cos \beta, \cos \alpha \sin \beta, -\sin \alpha) = -\frac{\partial e_r}{\partial \alpha} \sin \alpha \\
\frac{\partial e_r}{\partial \alpha} \times \frac{\partial e_r}{\partial \beta} &= \sin \alpha (\sin \alpha \cos \beta, \sin \alpha \sin \beta, \cos \alpha) = e_r \sin \alpha
\end{aligned}$$

We now differentiate $S = P + re_r$ with respect to α and β to obtain:

$$\begin{aligned}
\frac{\partial S}{\partial \alpha} &= \frac{\partial}{\partial \alpha}(P + re_r) = \frac{\partial r}{\partial \alpha} e_r + r \frac{\partial e_r}{\partial \alpha} \\
\frac{\partial S}{\partial \beta} &= \frac{\partial}{\partial \beta}(P + re_r) = \frac{\partial r}{\partial \beta} e_r + r \frac{\partial e_r}{\partial \beta} \\
\frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta} &= \left(\frac{\partial r}{\partial \alpha} e_r + r \frac{\partial e_r}{\partial \alpha} \right) \times \left(\frac{\partial r}{\partial \beta} e_r + r \frac{\partial e_r}{\partial \beta} \right) \\
&= r \frac{\partial r}{\partial \alpha} \left(e_r \times \frac{\partial e_r}{\partial \beta} \right) + r \frac{\partial r}{\partial \beta} \left(\frac{\partial e_r}{\partial \alpha} \times e_r \right) + r^2 \left(\frac{\partial e_r}{\partial \alpha} \times \frac{\partial e_r}{\partial \beta} \right) \\
&= \underbrace{-r \frac{\partial r}{\partial \alpha} \sin \alpha \left(\frac{\partial e_r}{\partial \alpha} \right) - r \frac{\partial r}{\partial \beta} \left(e_r \times \frac{\partial e_r}{\partial \alpha} \right)}_{\text{orthogonal to } e_r} + r^2 \sin \alpha (e_r)
\end{aligned}$$

Since we have a local parameterization of S in terms of α and β , we can write the outward unit normal to S as

$$N = -\frac{\frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta}}{\left\| \frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta} \right\|}$$

We also have $e_r \cdot N = -\cos \theta$ from the definition of θ . This allows us to write

$$\cos \theta \left\| \frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta} \right\| = e_r \cdot \left(\frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta} \right) = r^2 \sin \alpha$$

\Downarrow

$$\left\| \frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta} \right\| = \frac{r^2 \sin \alpha}{\cos \theta}$$

In terms of the spherical angles α and β , the solid angle element $d\alpha$ is given by

$$d\alpha = \sin \alpha \, d\alpha \, d\beta$$

which yields the change of variable formula

$$dS(X) = \left\| \frac{\partial S}{\partial \alpha} \times \frac{\partial S}{\partial \beta} \right\| d\alpha \, d\beta = \frac{r^2 \sin \alpha}{\cos \theta} d\alpha \, d\beta = \frac{r^2}{\cos \theta} d\alpha$$

which can be used in the following way:

$$\int_{\Gamma} \circledast d\alpha \, dX = \int_{\Gamma} \circledast \frac{\cos \theta}{r^2} dS \, dX$$

where \circledast denotes an arbitrary expression in the integrand. This gives the formula to go between focal plane coordinates and mixed coordinates:

$$d\alpha \, dX = \frac{\cos \theta}{r^2} dS \, dX \tag{2.9}$$

Surface coordinates (s_1, s_2, θ, ϕ) to mixed coordinates (s_1, s_2, x_1, x_2)

For a fixed point $S(s_1, s_2)$ on the surface, if the elevation angle θ of the ray unit vector e_r with respect to the surface unit normal N is varied, we obtain a local parameterization of the focal plane in terms of θ and the ϕ . This is essentially the same situation as just discussed but with a different reference frame. Therefore, we can obtain matching formulas

by swapping corresponding variables: $P \rightleftharpoons S$, $N \rightleftharpoons e_z$, $e_r \rightleftharpoons -e_r$, $\theta \rightleftharpoons \alpha$, $\phi \rightleftharpoons \beta$, $dS \rightleftharpoons dX$, and $d\boldsymbol{\theta} \rightleftharpoons d\boldsymbol{\alpha}$. From this we obtain the following parameterization speed (for the focal plane), solid angle element (from the point of view of the surface), and change of variable formula:

$$\begin{aligned} \left\| \frac{\partial P}{\partial \theta} \times \frac{\partial P}{\partial \phi} \right\| &= \frac{r^2 \sin \theta}{\cos \alpha} \\ d\boldsymbol{\theta} &= \sin \theta \, d\theta \, d\phi \\ dX(S) &= \left\| \frac{\partial P}{\partial \theta} \times \frac{\partial P}{\partial \phi} \right\| d\theta \, d\phi = \frac{r^2 \sin \theta}{\cos \alpha} d\theta \, d\phi = \frac{r^2}{\cos \alpha} d\boldsymbol{\theta} \end{aligned}$$

If we again let \otimes denote an arbitrary expression in the integrand, then the change of variable formula can be used in the following way:

$$\int_{\Gamma} \otimes d\boldsymbol{\theta} \, dS = \int_{\Gamma} \otimes \frac{\cos \alpha}{r^2} dX \, dS$$

This gives the formula to go between surface coordinates and mixed coordinates:

$$d\boldsymbol{\theta} \, dS = \frac{\cos \alpha}{r^2} dX \, dS \quad (2.10)$$

Focal plane coordinates $(x_1, x_2, \alpha, \beta)$ to surface coordinates (s_1, s_2, θ, ϕ)

We can now combine (2.9) and (2.10) to obtain the desired change of measure:

$$d\boldsymbol{\alpha} \, dX = \left(\frac{\cos \theta}{\cos \alpha} \right) d\boldsymbol{\theta} \, dS \quad (2.11)$$

Note that this result matches (2.8) from the first method as expected. It also matches (C.5b) and (C.10b) as obtained in the 2D case from the full Jacobian matrices in Appendix C.1.

2.3 Radiometry and the Image Irradiance Equation

It is now necessary to define the relevant radiometric quantities. We denote by $L(s_1, s_2, \theta)$ the surface radiance which, when multiplied by $\cos \theta$, is the power density emitted from the surface point $S(s_1, s_2)$ in the direction θ with respect to the surface normal per unit surface area dS per unit solid angle $d\theta$. Then the total radiant power \mathcal{P} emitted through the detected light rays Γ is given by

$$\mathcal{P} = \int_{\Gamma} L(s_1, s_2, \theta) \cos \theta dS d\theta \quad (2.12)$$

Next, the conjugate irradiance $E(X)$ is the incident power density from the detected light rays Γ at a point $P(X)$ on the focal plane per unit area dX . Thus, with Ω denoting the region of interest within the focal plane, called the conjugate image domain, the total conjugate irradiant power \mathcal{Q} passing through Ω is given by

$$\mathcal{Q} = \int_{\Omega} E(X) dX \quad (2.13)$$

Finally, the image irradiance $E'(X')$ is the incident power density from the detected light rays Γ at a point $P'(X')$ on the image plane per unit area dX' . Thus, with Ω' denoting the region of interest within the image plane, called the image domain (e.g. the region covered by the camera's image sensor), the total image irradiant power \mathcal{Q}' passing through Ω' is given by

$$\mathcal{Q}' = \int_{\Omega'} E'(X') dX' \quad (2.14)$$

Assuming that there is no power loss from the lens, power conservation requires that \mathcal{Q} and \mathcal{Q}' be equal, yielding

$$\int_{\Omega} E(X) dX = \int_{\Omega'} E'(X') dX' = \int_{\Omega} E'(X') M^2 dX \quad (2.15)$$

where (2.2b) is used to write both integrals over the conjugate image domain. From (2.15) it can be seen that the image irradiance is simply a scaled version of the conjugate image irradiance:

$$E'(X') = \frac{1}{M^2} E(X) \quad (2.16)$$

Furthermore, power conservation also requires that \mathcal{P} and \mathcal{Q} be equal, and making use of (2.11) we obtain

$$\begin{aligned} \int_{\Omega} E(X) dX &= \int_{\Gamma} L(s_1, s_2, \theta) \cos \theta dS d\theta \\ &= \int_{\Gamma} L(s_1, s_2, \theta) \cos \theta \left(\frac{\cos \alpha}{\cos \theta} \right) dX d\alpha \\ &= \int_{\Gamma} L(s_1, s_2, \theta) \cos \alpha d\alpha dX \\ &= \int_{\Omega} \left(\int_{\Gamma(X)} L(s_1, s_2, \theta) \cos \alpha d\alpha \right) dX \end{aligned}$$

Here $\Gamma(X) \subset \Gamma$ is the subset of the detected light rays that pass through the point $P(X)$ on the focal plane. The final step of the above yields the *conjugate irradiance equation*

$$E(X) = \int_{\Gamma(X)} L(s_1, s_2, \theta) \cos \alpha d\alpha \quad (2.17)$$

Thus finding the image irradiance can be done by computing the conjugate irradiance via (2.17) and then inputting the result into (2.16), giving the *image irradiance equation*

$$E'(X') = \frac{1}{M^2} \int_{\Gamma(X)} L(s_1, s_2, \theta) \cos \alpha d\alpha \quad (2.18)$$

where X and X' are related via (2.2a). It should be noted that (2.17) is identical to the rendering equation in computer graphics with the assumption of a unit reflectance map. This makes sense as our derivation models all radiance as being directly emitted from the

surface, not reflected. In preparation for Chapter 3, we can augment (2.18) by considering light emitted from not only the surface but also the background (e.g. the entirety of the visible scene that is not the surface-of-interest). Here we take the “blue sky” approach of [53] and model the background as another surface that fills the entire field of view. This background surface is represented by a sphere G of infinite radius that is parameterized by the spherical angular coordinates (η, γ) and supports a background radiance function K . Then the full image irradiance equation becomes

$$E'(X') = \frac{1}{M^2} \int_{\Gamma_S(X)} L(s_1, s_2, \theta) \cos \alpha d\alpha + \frac{1}{M^2} \int_{\Gamma_G(X)} K(\eta, \gamma, \theta_G) \cos \alpha d\alpha \quad (2.19)$$

where $\Gamma_S(X) \subseteq \Gamma(X)$ and $\Gamma_G(X) \subseteq \Gamma(X)$ are the subsets of the detected ray directions that pass through $P(X)$ that are emitted from the only surface and only the background, respectively, and θ_G denotes the ray angle with respect to the normal of the background G . Here, both integrals are dependent on the surface shape. But we can rewrite this as one integral dependent on the surface shape and one integral dependent only on the lens geometry by exploiting that fact that $\Gamma_S(X) \cup \Gamma_G(X) = \Gamma(X)$ and $\Gamma_S(X) \cap \Gamma_G(X) = \emptyset$:

$$\begin{aligned} E'(X') &= \frac{1}{M^2} \int_{\Gamma_S(X)} L(s_1, s_2, \theta) \cos \alpha d\alpha \\ &\quad + \frac{1}{M^2} \left(\underbrace{\int_{\Gamma(X)} K(\eta, \gamma, \theta_G) \cos \alpha d\alpha - \int_{\Gamma_S(X)} K(\eta, \gamma, \theta_G) \cos \alpha d\alpha}_{\int_{\Gamma_G(X)} K(\eta, \gamma, \theta_G) \cos \alpha d\alpha} \right) \\ &= \frac{1}{M^2} \int_{\Gamma_S(X)} (L(s_1, s_2, \theta) - K(\eta, \gamma, \theta_G)) \cos \alpha d\alpha + \frac{1}{M^2} \int_{\Gamma_G(X)} K(\eta, \gamma, \theta_G) \cos \alpha d\alpha \end{aligned} \quad (2.20)$$

The background can also be used to augment (2.17) in a similar manner by using (2.16):

$$E(X) = \int_{\Gamma_S(X)} (L(s_1, s_2, \theta) - K(\eta, \gamma, \theta_G)) \cos \alpha d\alpha + \int_{\Gamma(X)} K(\eta, \gamma, \theta_G) \cos \alpha d\alpha \quad (2.21)$$

2.3.1 Special Case: In-focus with Lambertian Reflectance (3D)

The integral in (2.17) has a closed-form solution in the special case of an in-focus point with Lambertian reflectance, which allows for two important simplifications. First, under Lambertian reflectance, the surface radiance is not a function of the ray angle, so $L(s_1, s_2, \theta) = L(s_1, s_2)$. Second, an in-focus surface point lies on the focal plane, so $S(s_1, s_2) = P(X)$, allowing L to be moved outside the integral as (s_1, s_2) will no longer depend on any angular property of the rays.

It will be easier to compute the integral over all allowable ray angles as opposed to over all allowable solid angles, so the solid angle element in (2.17) can be written in spherical coordinates as $d\alpha = \sin \alpha d\alpha d\beta$. For a point whose projection onto the plane of the lens lies inside the lens (e.g. $\|X\| < d/2$), the allowable azimuth ray angles span the entire interval $[-\pi, \pi]$. For points who project outside the lens, only a certain subinterval $[\beta_-, \beta_+]$ of azimuth angles are covered by the set of rays emitted from that point that pass through the lens. Similarly, for any given azimuth ray angle β , there exists an interval of valid elevation ray angles $[\alpha_-(\beta), \alpha_+(\beta)]$. Putting this together with the simplifications from the special case lets (2.17) be written as

$$E(X) = L(s_1, s_2) \int_{\beta_-}^{\beta_+} \int_{\alpha_-(\beta)}^{\alpha_+(\beta)} \cos \alpha \sin \alpha d\alpha d\beta \quad (2.22)$$

Let ρ designate the length of the projection of a ray onto the focal plane and $\rho_0 = \sqrt{x_1^2 + x_2^2}$ denote the radial distance from the optical axis for the specific in-focus point $P(X)$. Noting that $\tan \alpha = \rho/z$, the inner integral over α can be computed as

$$\begin{aligned}
\int_{\alpha_-(\beta)}^{\alpha_+(\beta)} \cos \alpha \sin \alpha d\alpha &= \frac{\tan^2 \alpha}{2(1 + \tan^2 \alpha)} \Big|_{\alpha_-(\beta)}^{\alpha_+(\beta)} \\
&= \frac{\rho^2}{2(\rho^2 + z^2)} \Big|_{\rho_-(\beta)}^{\rho_+(\beta)} \\
&= \frac{z^2(\rho_+^2 - \rho_-^2)}{2(\rho_+^2 + z^2)(\rho_-^2 + z^2)}
\end{aligned} \tag{2.23}$$

Note that whereas ρ_{\pm} , like α_{\pm} , are dependent on β , this dependence has been dropped from the notation for sake of neatness. What remains now is to find the form of this dependence of ρ_{\pm} on β . To do this, note that the two points in the plane of the lens $(x_1 + \rho_{\pm} \cos \beta, x_2 + \rho_{\pm} \sin \beta)$ both lie at a distance $d/2$ from the lens center, which is also the origin for this coordinate system. Thus we have the following quadratic expression for ρ_{\pm} :

$$\begin{aligned}
\left(\frac{d}{2}\right)^2 &= \|(x_1 + \rho_{\pm} \cos \beta, x_2 + \rho_{\pm} \sin \beta)\|_2^2 \\
&= x_1^2 + x_2^2 + \rho_{\pm}^2 + 2\rho_{\pm}(x_1 \cos \beta + x_2 \sin \beta) \\
&= \rho_0^2 + \rho_{\pm}^2 - 2\rho_{\pm}\rho_0 \cos(\beta - \beta_0)
\end{aligned} \tag{2.24}$$

where $\phi_0 = \arctan(x_2/x_1)$. Letting $\Delta\beta = \beta - \beta_0$ and solving (2.24) for ρ_{\pm} yields

$$\begin{aligned}
\rho_{\pm} &= \rho_0 \cos(\Delta\beta) \pm \sqrt{\rho^2 \cos^2(\Delta\beta) - \rho_0^2 + \left(\frac{d}{2}\right)^2} \\
&= \rho_0 \cos(\Delta\beta) \pm \sqrt{\left(\frac{d}{2}\right)^2 - \rho_0^2 \sin^2(\Delta\beta)}
\end{aligned} \tag{2.25a}$$

$$\begin{aligned}
\rho_{\pm}^2 &= \rho_0^2 (\cos^2(\Delta\beta) - \sin^2(\Delta\beta)) + \left(\frac{d}{2}\right)^2 \pm 2\rho_0 \cos(\Delta\beta) \sqrt{\left(\frac{d}{2}\right)^2 - \rho_0^2 \sin^2(\Delta\beta)} \\
&= \underbrace{\rho_0^2 \cos(2\Delta\beta)}_a + \underbrace{\left(\frac{d}{2}\right)^2 \pm 2\rho_0 \cos(\Delta\beta) \sqrt{\left(\frac{d}{2}\right)^2 - \rho_0^2 \sin^2(\Delta\beta)}}_b
\end{aligned} \tag{2.25b}$$

For the sake of space, note that (2.25b) is in the form $a \pm b$. We can now substitute (2.23) and (2.25b) into (2.22) to yield

$$\begin{aligned}
E(X) &= \frac{L(s_1, s_2)}{2} \int_{\beta_-}^{\beta_+} \frac{z^2((a+b) - (a-b))}{(a+b+z^2)(a-b+z^2)} d\beta \\
&= L(s_1, s_2) \int_{\beta_-}^{\beta_+} \frac{z^2 b}{(z^2 + a)^2 - b^2} d\beta \\
&= L(s_1, s_2) \int_{\beta_-}^{\beta_+} \frac{z^2 \rho_0 \sqrt{d^2 \cos^2(\Delta\beta) - \rho_0^2 \sin^2(2\Delta\beta)}}{\left(z^2 + \rho_0^2 \cos(2\Delta\beta) + \left(\frac{d}{2}\right)^2\right)^2 - d^2 \rho_0^2 \cos^2(\Delta\beta) + \rho_0^4 \sin^2(2\Delta\beta)} d\beta \\
&= L(s_1, s_2) \int_{\beta_-}^{\beta_+} \frac{z^2 \rho_0 \cos(\Delta\beta) \sqrt{d^2 - 4\rho_0^2 \sin^2(\Delta\beta)}}{\left(\rho_0^2 - \left(\frac{d}{2}\right)^2 - z^2\right)^2 + 4\rho_0^2 z^2 \cos^2(\Delta\beta)} d\beta \tag{2.26}
\end{aligned}$$

Note that the last equality is obtained using the identity $\cos(2\Delta\beta) = 2\cos^2(\Delta\beta) - 1$.

Next, a succession of variable changes are applied to (2.26) to simplify the integral into a cleaner and more easily-computable form:

$$\begin{aligned}
E(X) &= L(s_1, s_2) \int_{-\sin(\Delta\beta)}^{\sin(\Delta\beta)} \frac{z^2 \rho_0 \sqrt{d^2 - 4\rho_0^2 u^2}}{\left(\rho_0^2 - \left(\frac{d}{2}\right)^2 - z^2\right)^2 + 4\rho_0^2 z^2 (1 - u^2)} du \\
&\quad \left[u = \sin(\Delta\beta), \quad du = \cos(\Delta\beta) d\beta \right] \\
&= L(s_1, s_2) \int_0^{\frac{d}{2\rho_0}} \frac{2\rho_0 z \sqrt{z^2 d^2 - (2\rho_0 z u)^2}}{z^2 d^2 + \left(\rho_0^2 - \left(\frac{d}{2}\right)^2 + z^2\right)^2 - (2\rho_0 z u)^2} du \\
&= L(s_1, s_2) \int_0^A \frac{\sqrt{A^2 - v^2}}{A^2 + B^2 - v^2} dv, \quad \left[\begin{array}{ll} A = zd, & B = \rho_0^2 - \left(\frac{d}{2}\right)^2 + z^2, \\ v = 2\rho_0 z u, & dv = 2\rho_0 z du \end{array} \right] \\
&= L(s_1, s_2) \int_0^\infty \frac{1}{A^2} \frac{(A^2 - v^2)^2}{A^2 + B^2 - v^2} dw, \quad \left[w = \frac{v}{\sqrt{A^2 - v^2}}, \quad dw = \frac{A^2}{(\sqrt{A^2 - v^2})^3} dv \right] \\
&= L(s_1, s_2) = \int_0^\infty \frac{A^2}{B^2} \frac{A}{\left(1 + \frac{A^2}{B^2} + w^2\right)(1 + w^2)} dw, \quad \left[A^2 - v^2 = \frac{A^2}{1 + w^2} \right] \\
&= L(s_1, s_2) \int_0^\infty \left(\frac{1}{1 + w^2} - \frac{1}{1 + \frac{A^2}{B^2} + w^2} \right) dw
\end{aligned}$$

The final line of the above can be evaluated as

$$\begin{aligned} E(X) &= L(s_1, s_2) \left[\tan^{-1} w - \frac{1}{\sqrt{1 + \frac{A^2}{B^2}}} \tan^{-1} \left(\frac{w}{\sqrt{1 + \frac{A^2}{B^2}}} \right) \right]_0^\infty \\ &= \frac{\pi}{2} L(s_1, s_2) \left(1 - \frac{B}{\sqrt{A^2 + B^2}} \right) \end{aligned}$$

Substituting in for A and B yields the final closed-form solution to (2.22):

$$E(X) = \frac{\pi}{2} L(s_1, s_2) \left(1 - \frac{\rho_0^2 - \left(\frac{d}{2}\right)^2 + z^2}{\sqrt{z^2 d^2 + \left(\rho_0^2 - \left(\frac{d}{2}\right)^2 + z^2\right)^2}} \right) \quad (2.27)$$

Substituting (2.27) into (2.16) yields the final image irradiance

$$E'(X') = \frac{\pi}{2} \frac{1}{M^2} L(s_1, s_2) \left(1 - \frac{\rho_0^2 - \left(\frac{d}{2}\right)^2 + z^2}{\sqrt{z^2 d^2 + \left(\rho_0^2 - \left(\frac{d}{2}\right)^2 + z^2\right)^2}} \right) \quad (2.28)$$

The closed-form solution (2.28) has an analagous expression in the 2D case, which is derived and discussed in Appendix C.2.

2.4 Comparison to Horn's Equation

The classical irradiance equation is derived by Horn in [102]. The geometry considered is shown in Figure 2.6 where a lens of diameter d and focal length f is situated such that a surface patch of area δO is at depth z in front of the lens and the corresponding patch on the image plane, of area δI , is at depth f behind the lens.

By equating the solid angles of the cone of rays connecting these patches to the optical center and finding the ratio between the areas of the patches, it is found that the image irradiance E' is related to the surface radiance L by:

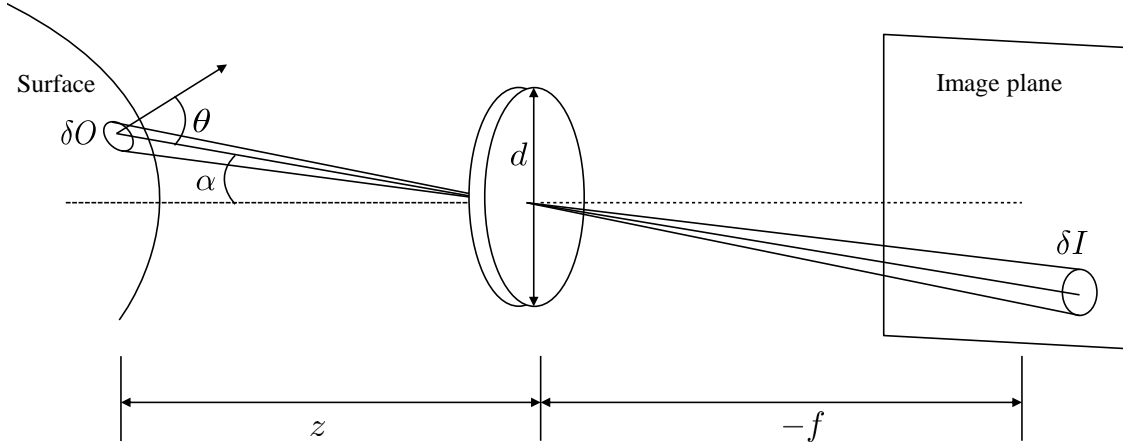


Figure 2.6: Geometry used by Horn to relate surface radiance to image irradiance. Only rays through the lens center are considered. Adapted with permission from [100] © 2017 IEEE

$$E'(X') = \frac{\pi}{4} \left(\frac{d}{f} \right)^2 L(s_1, s_2) \cos^4 \alpha \quad (2.29)$$

where α is again the angle that the ray connecting the centers of the two patches makes with the optical axis. For the discussion that follows, the term *Horn relationship* shall refer to (2.29) whereas the term *lens relationship* shall refer to (2.28).

2.4.1 Analytical Comparison

The reader is referred to [102] for the full derivation of (2.29). However, there are two key assumptions that are important to note. First, it is assumed that the only light rays arriving at the image patch are emitted from the surface patch. This is only true if the entire surface patch is perfectly focused, which brings up the second assumption. The image plane being placed a focal length behind the lens means that the lens is focused at infinity according to (2.1). In order for the surface patch to be perfectly focused, either z must be sufficiently large such that far-field approximations are valid, allowing the surface to be considered “infinitely” far from the lens, or a pinhole model must be implicitly assumed so that any point will be considered perfectly focused onto the image plane. If neither of these are true,

then some rays from nearby surface patches would arrive at the same image patch due to blur caused by defocus, causing the first assumption to be false. Also, the case of near-focused images is not addressed by either situation. This hints that the Horn relationship may be a special case of the lens relationship in far-field or small aperture conditions, and we shall now show that this is exactly the case.

It is easily noticed that the forms of (2.29) and (2.28) are very similar. They both involve the multiplication of a pure constant (a fraction of π), a constant dependent on the lens parameters, the surface radiance, and a spatially-varying term. Both equations depend on lens parameters d and f , but in the lens relationship there is no explicit dependence on ray angle α . Instead, the only other information needed to compute the image irradiance via (2.28) is the surface radiance, the placement of the image plane, and the location of the point being imaged, all of which should already be available without any additional computation. It should be noted that (2.29) can also be rewritten in a form that does not explicitly reference α by using the fact that $\cos \alpha = \frac{z}{\sqrt{z^2 + \rho_0^2}}$.

Figure 2.7 gives visual representations in the form of surface plots of (2.29) subtracted from (2.28) for varying values of d , z , and ρ_0 . In order to get a better understanding of the differing parts of the two irradiance relationships, the radiance has been set to $L = 1$. It was noticed that the deviations between the two relationships depended much more on the f -number of the lens (which is the ratio f/d of the lens' focal length to its diameter) than the actual value of the focal length, so for these plots a fixed focal length of $f = 50\text{mm}$ was used, which is the standard focal length for many modern cameras. The focus depth z was varied from 50mm to 1m as a lens does not create real images when focused at a depth less than its focal length, and this range allowed for exploration of both near-field and far-field conditions (relative to the focal length). On the other hand, the radial distance ρ_0 is only constrained by the size of the image sensor and was varied from 0m to 1m; though the size of image sensors rarely ever approach 1m in any dimension, this allows us to test for possible edge cases.

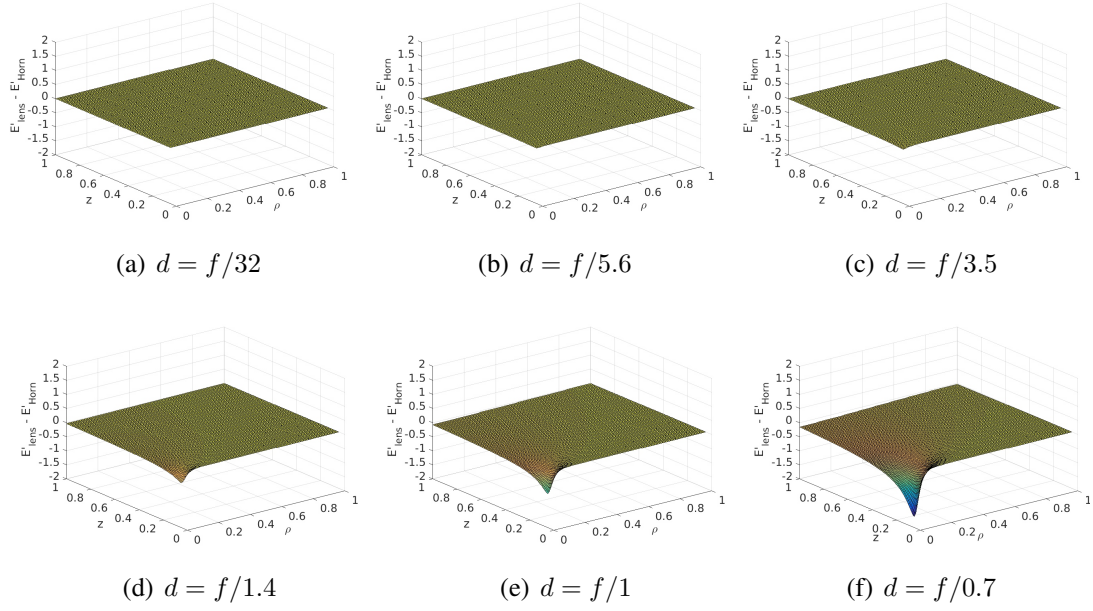


Figure 2.7: The Horn and lens irradiance equations are near identical for small apertures and far-focused points but diverge for near-focused and centered points as the aperture increases. This becomes significant for f -numbers less than about 2. Reused with permission from [101] ©2020 The Optical Society

Figure 2.7(a) shows the case of a small lens with diameter $d = f/32$ (this is typically the smallest aperture used in most general photography). It is seen that the surface plot is flat and near-constant no matter the values of z and ρ_0 . The maximum deviation is on the order of 10^{-4} when $z \approx f$ and $\rho_0 = 0$, and the deviation decreases as both z and ρ increase (order of 10^{-5} at $z = 20f$, for instance).

Next, Figures 2.7(b)-2.7(c) show the case of medium-sized lenses with diameters $d = f/5.6$ and $d = f/3.5$, which are standard lens sizes that comes packaged with many consumer-grade Digital Single Lens Reflex (DSLR) cameras. Again, the surface plot is visually flat except when both z and ρ_0 are low, where a small dip can be seen. This is also seen numerically, as now the maximum deviation is slightly higher. It is on the order of 10^{-2} when $z \approx f$ and $\rho_0 = 0$ for both f -numbers and on the order of 10^{-4} (for $f/5.6$) and 10^{-3} (for $f/3.5$) at $z = 20f$.

Finally, Figures 2.7(d)-2.7(f) show the case of large lenses with diameters $d = f/1.4$,

$d = f/1$, and $d = f/0.7$; it should be noted that 0.7 is the f -number of the largest practical lens ever made, so it is rare to find such low f -numbers outside of very specialized uses. A large dip can now be seen in the surface plots for these lens when both z and ρ_0 are small. In fact, for $f/0.7$ the maximum deviation has significantly increased to 1.6 when $z \approx f$ and only decreases to about 10^{-1} when $z = 20f$. For the other two slightly smaller lenses, the deviation is still large on the order of 10^{-1} at $z \approx f$ and 10^{-2} for $z = 20f$. This makes sense as the pinhole assumption used to derive (2.29) breaks down as the lens gets larger and farther away from a “pinhole.” It is important to note that as $L = 1$ in the generation of Figure 2.7, the actual deviation in modeled irradiance can be much larger if the scene lighting is bright.

The small deviation between these two irradiance relations in most situations is no coincidence. It can be shown that (2.29) is an approximation of (2.28) under two simple assumptions:

$$A1 : \rho_0^2 \ll z^2 \tag{2.30a}$$

$$A2 : d \ll z \tag{2.30b}$$

$A1$ ensures ray elevation angles are small and $\cos \alpha = \frac{z}{\sqrt{z^2 + \rho_0^2}} \approx 1$ whereas $A2$ ensures far-field conditions. Taking these two assumptions into account and writing the lens magnification as $M = z'/z$ yields

$$\begin{aligned}
E'(X') &= \frac{\pi}{2} L(s_1, s_2) \left(\frac{z}{z'} \right)^2 \left(1 - \frac{\rho_0^2 - \left(\frac{d}{2} \right)^2 + z^2}{\sqrt{z^2 d^2 + \left(\rho_0^2 - \left(\frac{d}{2} \right)^2 + z^2 \right)^2}} \right) \\
&\stackrel{A1}{\approx} \frac{\pi}{2} L(s_1, s_2) \left(\frac{z}{z'} \right)^2 \left(1 - \frac{-\left(\frac{d}{2} \right)^2 + z^2}{\sqrt{z^2 d^2 + \left(-\left(\frac{d}{2} \right)^2 + z^2 \right)^2}} \right) \\
&= \frac{\pi}{2} L(s_1, s_2) \left(\frac{z}{z'} \right)^2 \left(1 - \frac{-\left(\frac{d}{2} \right)^2 + z^2}{z^2 + \left(\frac{d}{2} \right)^2} \right) \\
&\stackrel{A2}{\approx} \frac{\pi}{4} L(s_1, s_2) \left(\frac{z}{z'} \right)^2 \left(\frac{d^2}{z^2} \right) \\
&= \frac{\pi}{4} L(s_1, s_2) \left(\frac{d}{z'} \right)^2 \tag{2.31}
\end{aligned}$$

Note that if $z' = f$, as is the case in Horn's derivation, then (2.31) is simply (2.29), assuming A1. If indeed $z' = f$, then by (2.1) the focal plane exists at infinity, meaning that both A1 and A2 hold for any finite choice of ρ and d .

2.4.2 Experimental Comparison

Whereas it is seen from the above that there are certain situations where the classical and proposed relationships give significantly different irradiance values, this alone does not indicate which of the two more closely models irradiance in the real world. The relative accuracy of the two equations were thus tested both with computer simulations using the optical design software Zemax as well as comparisons with a series of simple and intensity-sparse photographs. Finally, the accuracy of the the two irradiance models were compared in a simple application to shape-from shading.

Computer Simulations

Using Zemax's OpticStudio Online software, a simple optical geometry was set up and simulated as follows. The geometry consisted of three parts, very reminiscent of Figure 2.6: a lens, a small circular light source in front of the lens, and a rectangular light detector behind the lens. The lens was set to have a focal length of $f = 50\text{mm}$, and its diameter was set to one of three values: $d \in \{f/32, f/5.6, f/0.7\}$. These diameters were chosen as to be consistent with those used to generate Figure 2.7. The light source had a radius of $1\mu\text{m}$ and radiated a power of 1W isotropically within the solid angle subtended by the lens. This allowed the source to have a near-constant radiance and be considered a good approximation of a Lambertian point source. For each lens diameter, the source was placed at various depths $z \in \{1.1f, 10f, 20f\}$ and various radial distances $\rho \in \{0, d/4, 4d\}$; as the geometry was radially symmetric, it was sufficient to move off the optical axis in only the x -direction. This allowed for the analysis of near-focused and far-focused points as well as points on the optical axis, points that project inside the lens, and points that project outside the lens. For each depth of the source, the detector was placed at the appropriate depth z' given by (2.1) and radial position as specified by the lens magnification (2.2a) and measured the irradiance of the image of the source, which was computed using non-sequential ray-tracing.

For the sake of space, Table 2.1 shows only the most relevant results of these simulations, namely the absolute irradiance errors $|E'_{\text{meas}} - E'_{\text{Horn}}|$ and $|E'_{\text{meas}} - E'_{\text{lens}}|$, where E'_{meas} , E'_{Horn} , and E'_{lens} are the measured irradiance and the predicted irradiances given by the Horn and lens relationships, respectively. In all cases, the lens relationship provided a more accurate irradiance prediction than the Horn relationship, and in 2/3 of the cases the increase in accuracy was at least an order of magnitude. In general, the difference in accuracy was the most pronounced for near-focused points on the optical axis and tended to decrease as the light source was moved back away from the lens and radially outward as well as when the f -number was increased; this agrees with the patterns seen in Figure 2.7.

Table 2.1: Zemax Simulation Irradiance Errors (W/M^2)

	$\rho = 0$		$\rho = d/4$		$\rho = 4d$	
	$ E'_{\text{meas}} - E'_{\text{Horn}} $	$ E'_{\text{meas}} - E'_{\text{lens}} $	$ E'_{\text{meas}} - E'_{\text{Horn}} $	$ E'_{\text{meas}} - E'_{\text{lens}} $	$ E'_{\text{meas}} - E'_{\text{Horn}} $	$ E'_{\text{meas}} - E'_{\text{lens}} $
$d = f/32$						
$z = 1.1f$	3.83×10^{11}	7.10×10^6	2.55×10^{11}	1.35×10^8	4.70×10^{10}	3.08×10^8
$z = 10f$	6.11×10^{12}	4.92×10^{10}	5.09×10^{12}	1.05×10^{12}	3.30×10^{12}	2.55×10^{12}
$z = 20f$	1.27×10^{13}	2.05×10^{11}	1.30×10^{13}	4.71×10^{12}	1.29×10^{13}	1.14×10^{13}
$d = f/5.6$						
$z = 1.1f$	3.84×10^{11}	4.70×10^6	2.56×10^{11}	1.25×10^8	2.46×10^{10}	6.90×10^6
$z = 10f$	6.05×10^{12}	5.86×10^9	5.05×10^{12}	1.01×10^{12}	3.26×10^{12}	2.52×10^{12}
$z = 20f$	1.24×10^{13}	3.92×10^{10}	1.29×10^{13}	4.63×10^{12}	1.52×10^{13}	1.37×10^{13}
$d = f/0.7$						
$z = 1.1f$	5.00×10^{11}	2.59×10^8	3.04×10^{11}	1.67×10^8	1.90×10^8	7.37×10^6
$z = 10f$	6.15×10^{12}	4.82×10^{10}	1.91×10^{13}	1.49×10^{13}	2.15×10^{12}	1.71×10^{12}
$z = 20f$	1.09×10^{13}	1.72×10^{12}	7.99×10^{13}	7.16×10^{13}	1.32×10^{13}	1.18×10^{13}

Zemax simulations confirm the increased accuracy in the proposed thin lens irradiance model over Horn's classical model and that the improvement in accuracy increases for near-field, on-axis points and larger apertures. Reused with permission from [101] © 2020 The Optical Society.

Physical Experiments

In order to see how the two irradiance relationships compare in practice, they were both used to model the irradiance of an actual image. Using a Nikon DSLR camera equipped with two AF-S NIKKOR lenses of different aperture sizes, 24 photos were taken of a small and in-focus light source consisting of a red LED (NTE30034) shining through a pinhole, with the LED at a different radial location in each photo. The smaller lens had a focal length of $f = 18\text{mm}$ and aperture of $d = f/3.5 = 5.1\text{mm}$ whereas the larger lens had a focal length of $f = 50\text{mm}$ and aperture of $d = f/1.4 = 35.7\text{mm}$. Both lenses were focused near their minimum focus distance: 169.3mm for the small lens and 356.7mm for the large lens. In order to preserve the actual irradiance as much as possible, only the raw camera sensor data was used. The pinhole was used to ensure only one pixel on the camera sensor was illuminated, as seen in Figure 2.8, allowing the LED to be modeled as a point source. Considering the LED's half-power angle of 30° , the pinhole also allowed the radiance to be approximately Lambertian and constant so long as the pinhole was pointed towards

the optical center for each photo. The Lambertian quality is needed for both irradiance relationships to be valid, and the constant radiance allows for the system unknowns to be modeled as a pair of constants, one multiplicative and one additive.



Figure 2.8: Cropped photo taken of LED point source focused onto one pixel. Reused with permission from [101] ©2020 The Optical Society.

Fitting the irradiance models to the photos is thus equivalent to finding the optimal values of these constant parameters, which can be done by minimizing the squared-norm error between the measured irradiance value for each photo and the corresponding modeled irradiance:

$$C_{1,i}^*, C_{2,i}^* = \arg \min_{C_1, C_2} \left\| I - C_1 \hat{E}'_i - C_2 \right\|_2^2 \quad (2.32)$$

where $i \in \{\text{Horn}, \text{lens}\}$, I is the vector of illuminated pixel intensities (one pixel per image), \hat{E}'_{Horn} is the vector of irradiance values modeled by the Horn relationship (2.29), and \hat{E}'_{lens} is the vector of irradiance values modeled by the lens relationship (2.28), the latter two both with $L(s_1, s_2) = 1$. This has the well-known solution:

$$C_{1,i}^* = \frac{\sigma_{I\hat{E}'_i}}{\sigma_{\hat{E}'_i}^2}, C_{2,i}^* = \mu_I - \frac{\sigma_{I\hat{E}'_i}}{\sigma_{\hat{E}'_i}^2} \mu_{\hat{E}'_i} \quad (2.33)$$

where $\sigma_{I\hat{E}'_i}$ is the sample covariance between I and \hat{E}'_i , $\sigma_{\hat{E}'_i}^2$ is the sample variance of \hat{E}'_i , μ_I is the mean value of I , and $\mu_{\hat{E}'_i}$ is the mean value of \hat{E}'_i .

Quantitative results from this experiment can be seen in Tables 2.2 and 2.3, where

$E'_i = C_{1,i}^* \hat{E}'_i + C_{2,i}^*$. For both lenses, the lens relationship produced a lower overall modeling error than the Horn relationship, and the difference in modeling error increased nearly threefold from the smaller lens to the larger lens. Also, the lens relationship fit the majority of the images better in both cases (13/24 for the small lens and 18/24 for the large lens). It should be noted that whereas the difference in modeling error is relatively low (and the optimal models near identical in terms of the irradiance values they produce), the difference between the modeling parameters is significant, especially for the multiplicative constant $C_{1,i}^*$, which represents an estimate of the surface radiance. For most computer vision purposes this will not matter much, but it is quite notable from a modeling standpoint or in the case of inverse problems. This is all consistent with the previous discussion on how both irradiance models are nearly identical except in a few extreme situations.

Table 2.2: Analysis of Model Fitting Parameters

Method	$C_{1,i}^*$	$C_{2,i}^*$	$\ I - C_{1,i}^* \hat{E}'_i - C_{2,i}^*\ _2^2$
<i>f</i> /3.5			
Horn	1.0429×10^5	-4522.7	1.0819499×10^7
Lens	1.3097×10^5	-4530.5	1.0819465×10^7
<i>f</i> /1.4			
Horn	2.8846×10^4	652.2722	1.0934127×10^7
Lens	3.6211×10^4	651.6485	1.0934022×10^7

The optimal representation of the radiance differed significantly between irradiance models even though modeling error was similar.

Reused with permission from [101] ©2020 The Optical Society.

Shape-from-Shading Simulation

In order to illustrate the usefulness of the lens irradiance relationship, a simple shape-from-shading problem was simulated. The simulation had a camera imaging a Lambertian slanted plane that was illuminated by a distant point source with uniform radiance L_{src} , where the center of the lens was the origin of the coordinate system. The plane was sit-

Table 2.3: Analysis of Accuracy Differences in Modeled Irradiance

Aperture	$\ I - E'_{\text{Horn}}\ _2^2 - \ I - E'_{\text{lens}}\ _2^2$	$\max E'_{\text{Horn}} - E'_{\text{lens}} $
$f/3.5$	34.09	0.0144
$f/1.4$	105.25	0.0672

The lens-based irradiance model provided a better fit than Horn's, and the accuracy gap was most pronounced for the large lens.

Reused with permission from [101] ©2020 The Optical Society.

uated such that its center was perfectly in-focus (e.g. it lay in the focal plane) with unit normal N . The image irradiance was computed using (2.28) (the more accurate of the two irradiance models), with the scene radiance computed as $L = L_{\text{src}} \cos \theta$, where θ is the angle between N and the ray vector between the point source and a given point on the plane. Assuming only the irradiance pattern and illumination are known, as well as that the surface is a plane, recovering the shape is equivalent to recovering the unit normal defining the plane. This was done using gradient descent on the squared-difference between the measured ray-cosines and those that result from an estimate of the unit normal, N_{est} . Since in both irradiance models the image irradiance is proportional to the scene radiance, the measured values of $\cos \theta$ can be computed by dividing the measured irradiance by the factors multiplying L and by L_{src} . This was done twelve times, comparing the results of using (2.28) versus (2.29) in the inversion process with three different lens apertures for both near-field and far-field imaging.

The results of these simulations can be seen in Table 2.4 and are summarized as follows, where $f = 50\text{mm}$, $z \in \{1\text{m}, 100\text{m}\}$, the initial normal estimate was $N_{\text{est}} = [0, 0, -1]^T$, $N = [1/2, 1/2, -1/\sqrt{2}]^T$, $L_{\text{src}} = 1\text{W/sr/m}^2$, and $d \in \{f/32, f/3.5, f/1\}$. The lens aperture had no apparent effect on the reconstructed normal, so the following results apply to all three apertures. Both models gave near-perfect reconstructions in the far-field and only mediocre reconstructions in the near-field, though the lens irradiance model provided a more accurate unit normal in both cases even if it provided a worse value of the objec-

Table 2.4: Shape-from-Shading Simulation Results

Method	N_{est}	$\ N - N_{\text{est}}\ _2$	$\sum (\cos \theta - \cos \theta_{\text{est}})^2$
$z = 1\text{m}$			
Horn	$[0.6519, 0.0426, -0.7571]^T$	4.86×10^{-1}	3.3832
Lens	$[0.6022, 0.2600, -0.7549]^T$	2.65×10^{-1}	5.0956
$z = 100\text{m}$			
Horn	$[0.4997, 0.5001, -0.7072]^T$	2.92×10^{-4}	0.0449
Lens	$[0.5000, 0.5000, -0.7071]^T$	1.62×10^{-5}	0.0082

The lens model reconstructed the unit normal more accurately than Horn's model.

The improvement was notable even in the far-field where the two models are near-identical. Reused with permission from [101] ©2020 The Optical Society.

tive function for the near-field. Given the known ambiguous nature of shape-from-shading, it is possible the near-field case contained a hard-to-escape local minimum even with all the assumptions used; this would explain the poorer performance of both models in the near-field.

CHAPTER 3

THIN LENS MODEL INVERSION

With the thin lens forward model established, we can set up a cost functional that is minimized when the surface shape and radiance estimates are optimal. Here optimal means that the reconstruction produces images (using the forward model (2.19)) as close as possible to the input images in the sense of the \mathcal{L}_2 -norm.

Since we will be estimating the surface shape S and the surface and background radiance functions L and K using an iterative procedure, we can augment them to be time-varying so that they represent a class of evolving functions $S(u, v) \rightarrow S(u, v, t)$, $L(u, v) \rightarrow L(u, v, t)$, and $K(\eta, \gamma) \rightarrow K(\eta, \gamma, t)$, where t is an artificial gradient descent time parameter and (u, v) are time-independent surface parameters. Letting C be the total number of images, we may construct the image error function \mathcal{E}'_c comparing the modeled image irradiance E'_c and the actual measured intensity I_c of the c -th image as

$$\mathcal{E}'_c(X'_c) = E'_c(X'_c) - I_c(X'_c)$$

where the subscript on X' is used to denote that these planar coordinates are with respect to the coordinate system of the c -th image. This same error function can be mapped to the conjugate domain Ω_c to produce an equivalent conjugate error function

$$\mathcal{E}_c(X_c) = \mathcal{E}'_c(M_c X_c) = M_c^{-2} E'_c(X_c) - I_c(M_c X_c)$$

The total squared-error cost function J for the full image collection is then

$$J = \frac{1}{2} \sum_{c=1}^C \int_{\Omega'_c} (\mathcal{E}'_c)^2 dX'_c = \frac{1}{2} \sum_{c=1}^C \int_{\Omega_c} M_c^2 \mathcal{E}_c^2 dX_c \quad (3.1)$$

We plan on minimizing J through an artificial-time gradient descent procedure, so the derivative of J with respect to the artificial time parameter t is

$$\frac{dJ}{dt} = \sum_{c=1}^C \int_{\Omega_c} \mathcal{E}_c \frac{dE_c}{dt} dX_c \quad (3.2)$$

Note that (3.1) contains no artificial regularizers and only measures the image matching-error. If images are sufficiently defocused, there should be natural regularization that occurs through the use of the thin lens model. This is already hinted at in the form of (2.17), where the irradiance is a weighted average over all scene points that contribute to a respective image point, and will be even more evident in the evolution equations derived here.

We want to minimize (3.1) with respect to S , L , and K , and this can be done using an alternating gradient descent procedure. First, we fix an initial estimate for the surface shape S and find the optimal surface and background radiance functions L and K for this estimate. Then, we fix L and K and update S according to the gradient descent flow of J with respect to S . These two steps are then repeated until convergence. From (3.2) we can see that in order to find the sensitivity of J to perturbations in the surface shape and scene radiance, and thus obtain the desired gradient descent evolution equations, it suffices to find the corresponding sensitivities of the irradiance E . For the following, Lambertian reflectance is assumed, so the dependence of L on θ has been dropped in the notation.

3.1 Surface Evolution Equation

Here we develop the equations governing the desired evolution of S that minimize (3.1). For the sake of space and clarity of notation, the following development is done with respect to only a single image unless otherwise specified. However, it is straightforward to apply the results of this section to the entire image collection: for a given surface point, compute the evolution flow with respect to each image where that point is visible, and then sum

these individual updates for the full evolution flow.

3.1.1 Radial/transverse coordinate frame

For the following it will be useful to define the frame

$$N, \quad T_r = e_r - (e_r \cdot N)N, \quad T_\perp = T_r \times N \quad (3.3)$$

where T_r denotes e_r projected onto the tangent plane of S and T_\perp denotes the orthogonal transverse direction. Along the occluding boundary S^* they become unit vectors $T_r^* = e_r^*$ and $T_\perp^* = e_r^* \times N^*$ but are otherwise orthogonal with matching magnitude $\|T_r\| = \|T_\perp\| \leq 1$ and may therefore form the basis for isothermal coordinates (s_r, s_\perp) in the radial and transverse directions respectively (with unit speed along the occluding boundary). The $*$ superscript is used here and in the following to denote the corresponding quantity at the boundary of visibility. Note that these coordinates become degenerate at any front-to-parallel point where $e_r = -N$.

3.1.2 Irradiance sensitivity

Using (2.9) and noting that $\cos \theta = -e_r \cdot N$, we may rewrite the first term of (2.21) as a flux integral:

$$E(X) = - \int_{S(\Gamma_S(X))} \frac{\hat{L} - \hat{K} \cos \alpha}{r^2} e_r \cdot N \, dS + \int_{\Gamma(X)} K \cos \alpha \, d\alpha \quad (3.4)$$

where $\hat{L} : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $\hat{K} : \mathbb{R}^3 \rightarrow \mathbb{R}$ are volumetric extensions of L and K such that $\hat{L}(S) = L(S)$ and $\hat{K}(G) = K(G)$. If we assume that \hat{L} and \hat{K} are fixed and that only the surface evolves, then differentiating with respect to time causes the second integral to vanish, and the time derivative of (3.4) is given as

$$\begin{aligned} \frac{dE}{dt}(X) &= \int_{S_{(\Gamma_S(X))}^*} \left(-\frac{\hat{Q}^* \cos \alpha^*}{(r^*)^2} e_r^* \right) \cdot \left(\frac{dS^*}{dt} \times T^* \right) ds^* \\ &\quad + \int_{S_{(\Gamma_S(X))}} \nabla \cdot \left(-\frac{\hat{Q} \cos \alpha}{r^2} e_r \right) (S_t \cdot N) dS \end{aligned} \quad (3.5)$$

where $S_{(\Gamma_S(X))}^*$ (or more compactly S^*) denotes the surface curve bounding the subpatch of S visible from $P(X)$, T^* denotes the unit tangent vector along S^* (which will also be a unit tangent vector to the surface S), ds^* denotes the arclength element of S^* , and $\hat{Q} \doteq \hat{L} - \hat{K}$. A detailed derivation of (3.5) is given in Appendix A. Next, by noting that

$$\nabla \cdot \left(\frac{\hat{Q} \cos \alpha}{r^2} e_r \right) = \frac{\cos \alpha}{r^2} (\nabla \hat{Q} \cdot e_r)$$

we obtain

$$\begin{aligned} \frac{dE}{dt}(X) &= \int_{S_{(\Gamma_S(X))}^*} \left(-\frac{\hat{Q}^* \cos \alpha^*}{(r^*)^2} e_r^* \right) \cdot \left(\frac{dS^*}{dt} \times T^* \right) ds^* \\ &\quad - \int_{S_{(\Gamma_S(X))}} \nabla \hat{Q} \cdot e_r \frac{\cos \alpha}{r^2} (S_t \cdot N) dS \end{aligned} \quad (3.6)$$

Interior boundary points (non-occluding)

A boundary point S^* which does not fall along a visibility boundary with respect to $P(X)$ represents the endpoint of a ray along the boundary $\partial\Gamma_S(X)$ of the surface ray set between $P(X)$ and the lens. For such points, the ray direction at the boundary remains unchanged even as S^* evolves (e.g. S^* moves along the ray), and we have that

$$e_r^* = \frac{S^* - P}{r^*} = \text{constant} \quad \xrightarrow{\frac{d}{dt}} \quad \frac{dS^*}{dt} = \frac{dr^*}{dt} e_r^*$$

Such points do not contribute to the boundary integral in (3.6), since $e_r^* \cdot (e_r^* \times T^*) = 0$,

and may therefore be ignored when evaluating the boundary integral.

Occluding boundary points

We may then restrict our attention to the subset of $S^*_{(\Gamma_S(X))}$ representing occluding boundaries with respect to $P(X)$. As such, this subset of S^* is represented by the occluding boundary condition

$$e_r^* \cdot N^* = 0$$

We may decompose S_t into its tangential and normal evolution components using the radial/transverse frame (3.3)

$$S_t = a_r T_r + a_\perp T_\perp + b N$$

Now letting $s_r^*(t)$ and $s_\perp^*(t)$ denote evolving isothermal coordinates paired to T_r^* and T_\perp^* at an occluding boundary point S^* we may write

$$\frac{dS^*}{dt} = \left(a_r^* + \frac{ds_r^*}{dt} \right) T_r^* + \left(a_\perp^* + \frac{ds_\perp^*}{dt} \right) T_\perp^* + b^* N^*$$

and in turn

$$e_r^* \cdot \left(\frac{dS^*}{dt} \times T^* \right) = e_r^* \cdot (N^* \times T^*) (S_t^* \cdot N^*)$$

We can represent T^* in the radial/transverse frame as

$$T^* = - \frac{\tau_r^* T_r^* + \kappa_r^* T_\perp^*}{\sqrt{(\kappa_r^*)^2 + (\tau_r^*)^2}}$$

where κ_r^* is the radial curvature (e.g. the normal curvature of S in the radial direction T_r^*) and τ_r^* is the radial torsion (e.g. the geodesic torsion of S in the radial direction T_r^*). We can then write the scalar triple product

$$e_r^* \cdot (N^* \times T^*) = -\frac{\kappa_r^*}{\sqrt{(\kappa_r^*)^2 + (\tau_r^*)^2}}$$

This allows us to rewrite (3.6) as

$$\begin{aligned} \frac{dE}{dt}(X) &= \int_{S_{(\Gamma_S(X))}^*} \left(\frac{\hat{Q}^* \cos \alpha^*}{(r^*)^2} \right) \left(\frac{\kappa_r^*}{\sqrt{(\kappa_r^*)^2 + (\tau_r^*)^2}} \right) (S_t^* \cdot N^*) ds^* \\ &\quad - \int_{S_{(\Gamma_S(X))}} \nabla \hat{Q} \cdot e_r \frac{\cos \alpha}{r^2} (S_t \cdot N) dS \end{aligned} \quad (3.7)$$

3.1.3 Total matching error sensitivity

Inserting the occluding boundary term of (3.7) into (3.2) and swapping the order of integration (details in Appendix B) yields

$$\begin{aligned} &\int_{\Omega} \mathcal{E} \left(\int_{S_{(\Gamma_S(X))}^*} \left(\frac{\hat{Q}^* \cos \alpha^*}{(r^*)^2} \right) \left(\frac{\kappa_r^*}{\sqrt{(\kappa_r^*)^2 + (\tau_r^*)^2}} \right) (S_t^* \cdot N^*) ds^* \right) dX \\ &= \int_S \frac{\hat{Q}}{\sqrt{1 - (e_z \cdot N)^2}} \left(\int_{\Omega_{(\Gamma_S(S))}^*} \frac{\mathcal{E}^* \kappa_r^* \cos \alpha^*}{r^*} ds_X \right) (S_t \cdot N) dS \end{aligned} \quad (3.8)$$

where $\Omega_{(\Gamma_S(S))}^*$ is the line of points in the focal plane that make rays satisfying the occluding boundary condition with the surface point S , and ds_X is the arclength element in the focal plane.

Next, inserting the surface integral term of (3.7) into (3.2) and swapping the order of integration yields

$$\begin{aligned}
& \int_{\Omega} \mathcal{E} \left(- \int_{S(\Gamma_S(X))} \nabla \hat{Q} \cdot e_r \frac{\cos \alpha}{r^2} (S_t \cdot N) dS \right) dX \\
&= \int_S -\nabla \hat{Q} \cdot \left(\int_{\Omega(\Gamma_S(S))} \frac{\mathcal{E} \cos \alpha}{r^2} e_r dX \right) (S_t \cdot N) dS
\end{aligned} \tag{3.9}$$

Adding (3.8) and (3.9) and applying (2.10) gives the full expression for $\frac{dJ}{dt}$ when only S is evolving and allows us to extract the evolution equation for S for a single image as

$$S_t = \left[-\frac{\hat{Q}}{\sqrt{1 - (e_z \cdot N)^2}} \left(\int_{\Omega^*(\Gamma_S(S))} \frac{\mathcal{E}^* \kappa_r^* \cos \alpha^*}{r^*} ds_X \right) + \nabla \hat{Q} \cdot \left(\int_{\Gamma_S(S)} \mathcal{E} \cos \theta e_r d\theta \right) \right] N \tag{3.10}$$

Notice that each surface point is updated by a weighted averaging of the pointwise error \mathcal{E} over the region of the image contributed to by that surface point. This is the key that leads to the natural regularization mentioned previously, as the impact of noise or other outlying measurements in the data will be averaged out, with the size of the averaging window correlating with the amount of defocus present in the images. Also, in the special case of modeling the radiances L and K as constant functions, the second term vanishes, meaning that only surface points on the occluding boundary need to be considered in that case.

One final thing to note is that the first integral of (3.10) represents a diffusion term due to the presence of the radial curvature κ_r . Since both \hat{Q} and \mathcal{E} can have either positive or negative sign, this diffusion can be in the backwards direction and thus unstable. If the curvature term were outside the integral, we could simply set $\kappa_r = -1$ which would at least guarantee the update at each point is in the right direction since we know that κ_r is negative at occluding boundary points (since we are using an outward normal). But since κ_r is inside the integral, doing this normalization could potentially change the overall sign of the integral and thus the direction of the update. This can be avoided by keeping track of

the sign of the actual gradient and flipping the sign of the normalized update if necessary, as the computation of the radial curvature is relatively inexpensive. Such a modification turns the diffusion into an advection, which can be stabilized with the proper choice of time step. However, in doing so we are no longer descending down the exact gradient of the cost function J , though we are traversing a shallower trajectory that will still lead to its minimum.

Like with the irradiance equation, the analogous equation for (3.10) in the 2D case is derived and discussed in Appendix C.4. While the two cases give functionally similar evolution flows, one important difference is that the cosine factor present in the first term of (3.10) is completely canceled out in 2D. By expanding

$$\sqrt{1 - (N \cdot e_z)^2} = \sqrt{\cos^2 \alpha + ((e_z \times e_r) \cdot N)^2}$$

it is seen that only when N lies in the plane spanned by e_r and e_z does the above simplify to $\cos \alpha$, which in turn cancels out the cosine factor in the integral. This is the only possible scenario in 2D, but it is far from guaranteed in 3D, explaining both the new radical term and the non-cancellation of the cosine term in the 3D case.

3.2 Scene Radiance Estimation

3.2.1 Smooth radiances

Once the surface has been updated according to (3.10), we need to update L and K as to be optimal with this new shape. If we assume S and K are fixed and that only L is evolving, the sensitivity of E with respect to L is

$$\frac{dE}{dt}(X) = - \int_{S(\Gamma_S(X))} L_t \frac{\cos \alpha}{r^2} e_r \cdot N \, dS$$

The analogous expression for the sensitivity of E with respect to K for fixed S and L can be found by replacing $\Gamma_S \rightarrow \Gamma_G$ and $L_t \rightarrow K_t$. After inserting into (3.2), swapping the order

of integration, and applying the change of variables (2.10), the gradient descent evolution equations for L and K over the entire image collection are

$$L_t = \sum_{c=1}^C \int_{\Gamma_S(S)} \mathcal{E}_c \cos \theta d\theta \quad (3.11a)$$

$$K_t = \sum_{c=1}^C \int_{\Gamma_G(G)} \mathcal{E}_c \cos \theta_G d\theta_G \quad (3.11b)$$

where θ_G denotes the ray angle with respect to the normal of the background surface G and the background solid angle element $d\theta_G$.

In order to obtain the optimal radiance functions for a given estimate of S , L and K can be updated in an alternating fashion according to (3.11a) and (3.11b) until steady state is reached. The alternating update is necessary as \mathcal{E} is dependent on both L and K for some pixels through defocus blur, therefore coupling the update equations.

3.2.2 Constant radiances

In the special case that the surface and background radiance functions are separately modeled as constant functions, which is the case considered in the experiments in Chapter 4, then the optimal values of L and K individually can be found by finding where the derivative of J vanishes, yielding

$$L_{\text{opt}} = \frac{\sum_{c=1}^C \left(\int_{\Omega'_c} I_c(X'_c) W'_S(X'_c) - K W'_G(X'_c) W'_S(X'_c) dX'_c \right)}{\sum_{c=1}^C \int_{\Omega'_c} W'_S(X'_c)^2 dX'_c} \quad (3.12a)$$

$$K_{\text{opt}} = \frac{\sum_{c=1}^C \left(\int_{\Omega'_c} I_c(X'_c) W'_G(X'_c) - L W'_G(X'_c) W'_S(X'_c) dX'_c \right)}{\sum_{c=1}^C \int_{\Omega'_c} W'_G(X'_c)^2 dX'_c} \quad (3.12b)$$

where $W'_i(X_c) = M^{-2} \int_{\Gamma_i(X_c)} \cos \alpha \, d\alpha$ for $i \in \{S, G\}$. Note that (3.12a) and (3.12b) are still coupled, so they still need to be applied in an alternating fashion until steady state is reached. However, it was seen in practice that a good steady state approximation is usually reached after two applications despite the coupling. This constant radiance case is analagous to the assumptions used in Chan-Vese image segmentation, where the images are assumed to be well-approximated as binary images. This interpretation is still mostly valid with the thin lens model though for a larger class of images; the subtle difference here is that given sufficient defocus, there will exist at least one smooth transition region where the near-constant foreground and background blur and blend with each other.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

Here we show the results of testing our method on four different datasets, three computer-generated with Blender and one obtained with real cameras. All experiments were run on a laptop equipped with an Intel i7-4790 processor (3.60 GHz), assumed that the radiance functions L and K were constant functions, and used a level set implementation with a $128 \times 128 \times 128$ voxel grid. The dimensions (in pixels) of the Blender-generated images and real images were 480×480 and 2304×1536 respectively. For comparison purposes, we also applied the pinhole-based method of [53] to these datasets.

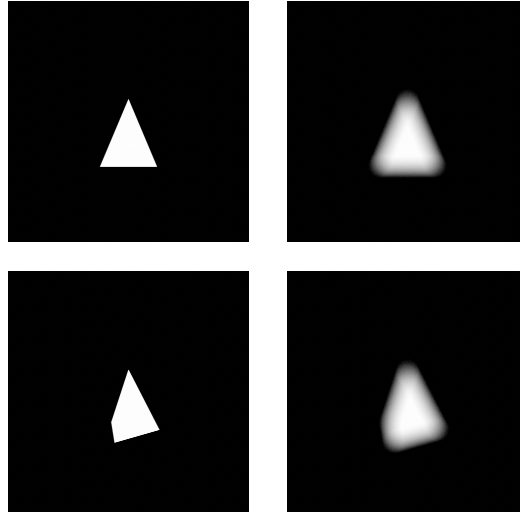


Figure 4.1: Focused (left) and defocused (right) images from the first tetrahedron dataset (2 of 30 views). Defocus blur makes sharp corners appear rounded and causes the tetrahedron to appear larger than its actual size

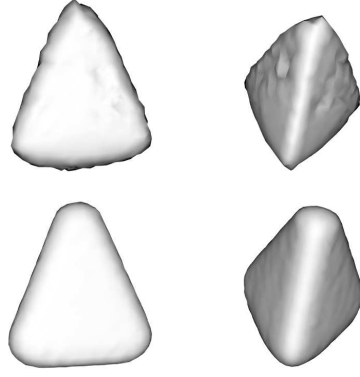


Figure 4.2: Two viewpoints of the thin lens-based reconstruction (top) and the pinhole-based reconstruction (bottom) for the first tetrahedron dataset. The sharp edges and corners of the shapes were reconstructed accurately using the thin lens method but rounded using the pinhole method

4.1 Qualitative Analysis

4.1.1 Synthetic Images

The first two datasets consist of a single tetrahedron imaged in two different ways: varying viewpoint only, and varying both viewpoint and focal parameters. For both datasets, thirty cameras were placed in a 20 m diameter ring around and slightly above the tetrahedron. For the first dataset, the cameras had identical intrinsic parameters of $f = 50$ mm, $d = f/1.4$, and $z = 500$ mm; the right column of Figure 4.1 shows two of the images produced by these cameras while the left column shows the same views of the scene when the cameras are focused. As would be expected, the tetrahedron appears larger in the defocused images due to defocus blur, and the edges and corners appear rounded and elongated instead of sharp. The initial surface used was an ellipsoid containing the tetrahedron.

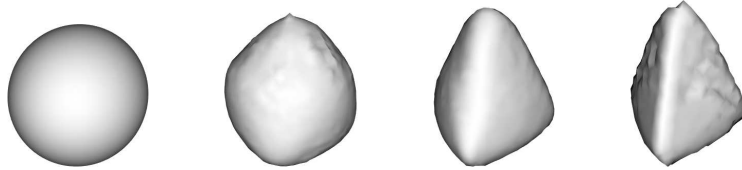


Figure 4.3: Four snapshots of the evolving surface for first tetrahedron dataset using the thin lens method after 0, 20, 60, and 120 iterations

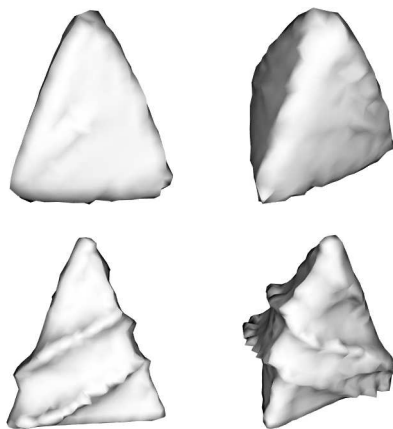


Figure 4.4: Two viewpoints of the thin lens-based reconstruction (top) and the pinhole-based reconstruction (bottom) for the second tetrahedron dataset. The thin lens method successfully reconstructed the single tetrahedron while the pinhole method could not reconcile the two different thicknesses visible in the images

Figure 4.2 shows two viewpoints of the reconstructed models obtained using both the thin lens method and pinhole method, while Figure 4.3 shows the evolving surface at four different iterations for the thin lens method, including the initial and final surface. Both models are the correct general shape, but the thin lens method was able to reconstruct the sharp edges and corners of the tetrahedron, particularly the edges between the visible faces, whereas the pinhole method obtained much wider and more rounded features. This is because the pinhole-based method incorrectly assumes perfect focus, so the blurred, rounded features in the image are assumed to be directly indicative of the scene shape instead of a consequence of the imaging process.

The differences between how the thin lens and pinhole methods treat defocus are shown even more dramatically in the second dataset. With the cameras placed as before, the focal parameters are varied such that the first 15 cameras had $f = 50$ mm, $d = f/1.4$, and $z = 2$ m while the latter 15 cameras had $f = 50$ mm, $d = f$, and $z = 500$ mm. Thus, the first 15 images appear relatively sharp and well-focused while the latter 15 appear much more blurred and defocused, similar respectively to the left and right columns of Figure 4.1. We see from Figure 4.4 that the pinhole method completely fails here and reconstructs the

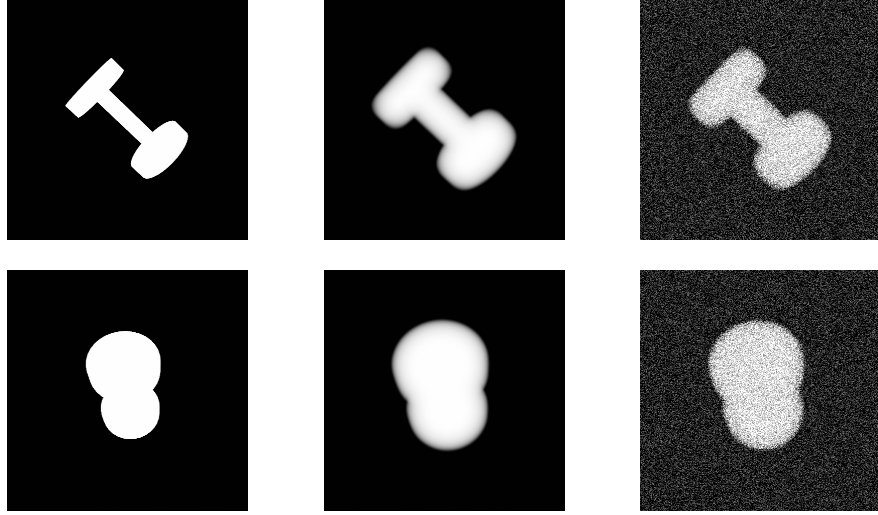


Figure 4.5: Focused (left), defocused (middle), and noisy and defocused (right) images from the dumbbell dataset (2 of 30 views). Defocus blur makes the entire object appear wider and the disk edges appear rounded instead of being a sharp transition

shape as a conglomeration of two tetrahedra of different thicknesses. This failure occurs because the pinhole model cannot reconcile the discrepancies between the two apparent sizes of the tetrahedron that can be seen in the images. In contrast, our thin lens method treats defocus as an imaging property, so it was able to account for the different levels of defocus blur and reconstruct the tetrahedron as a single coherent shape. It should be noted that the bottom face of the tetrahedron had a relatively rough texture in both the thin lens reconstructions, due to the non-visibility of this face in the images and the lack of the smoothness penalty used in the pinhole method. However, the three other faces are mostly smooth, showing that no smoothness penalty is needed for patches of the surface that are visible to the cameras.

To test our method’s performance on rounded and partially concave shapes, the third dataset consists of a slanted dumbbell (long cylindrical rod with flat disks on each end), with the cameras situated as before and with focal parameters matching those of the first tetrahedron dataset. The imaged scene is illustrated in Figure 4.5, with the left and middle columns respectively corresponding to focused images of the dumbbell and defocused images actually used for the reconstruction. Again, the initial surface was an ellipsoid con-

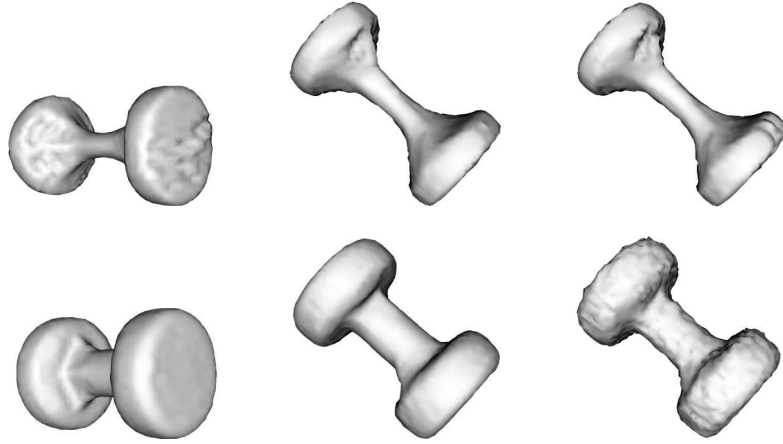


Figure 4.6: Two viewpoints (left and middle) of the thin lens-based reconstruction (top) and the pinhole-based reconstruction (bottom) for the dumbbell dataset. The thin lens method more accurately reconstructs the sharp edges of the disks and the actual thickness of both the disks and the central rod when compared with the pinhole method. When noise was added to the dumbbell dataset (right), the thin lens reconstruction was barely affected while the pinhole reconstruction was of noticeably lower quality

taining the dumbbell, and two views of the resulting reconstructions from each method are shown in Figure 4.6, with Figure 4.7 containing snapshots of the evolution process using the thin lens method. The thin lens reconstruction more accurately models the sharp transitions at the edges of the disks along with the actual thickness of the disks and rod compared to the pinhole reconstruction. It was also able to well-model the rounded surfaces and concave areas with only slightly less smoothness relative to the pinhole reconstruction. In contrast, the pinhole method once again rounded all the edges and produced a too-thick reconstruction, which can be easily seen in the relative thicknesses of the reconstructed rod and disks.

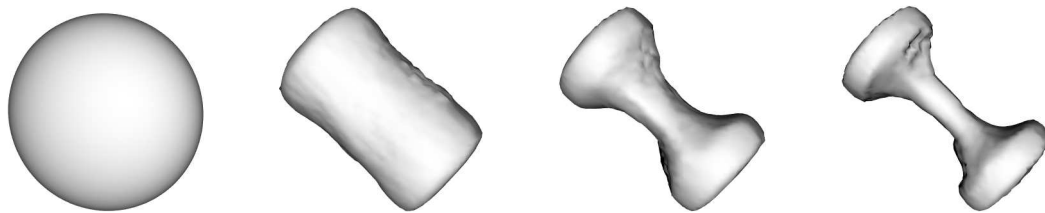


Figure 4.7: Four snapshots of the evolving surface for the dumbbell dataset using the thin lens method after 0, 160, 240, and 400 iterations

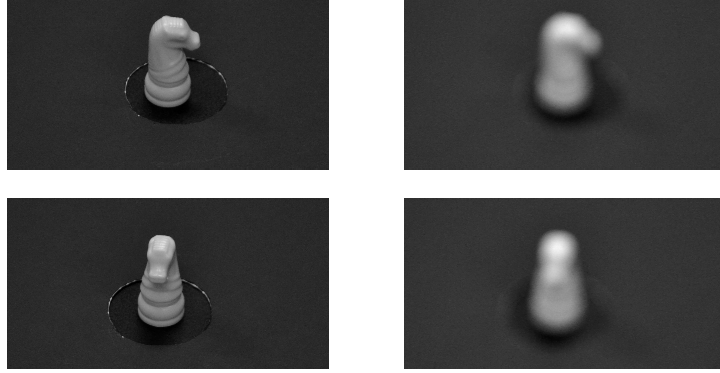


Figure 4.8: Zoomed-in focused (left) and defocused (right) images from the chesspiece dataset (2 of 32 views). Sharp features like occluding edges and texture on the mane are blurred out and indistinguishable in the defocused images.

To test robustness to noise, a second test was run after adding uniformly distributed noise to the dumbbell images; two of these noisy images can be seen in the right column of Figure 4.5. A one-view comparison of the resulting reconstructed models can be seen in the right column of Figure 4.6. While the noise barely affected the quality of the thin lens reconstruction, it caused a noticeable decrease in the smoothness of the pinhole reconstruction, especially on the rounded edges of the disks and central rod. This was expected due to the averaging that takes place in the surface update for the thin lens method but which is not present in the pinhole method.

4.1.2 Real Images

The fourth dataset consists of 32 real images of a white knight chess piece taken using a DSLR camera; though the camera was in a fixed position, the knight was placed on a turntable that was rotated in order to generate images at different viewpoints. Zoomed-in versions of two of these viewpoints can be seen in Figure 4.8, where as before the left column contains focused images while the right column contains defocused images that were used in the reconstruction. It can be seen that while the general shape of the knight is maintained in the defocused images, specific details like occluding edges and texture are blurred out and indistinguishable, particularly around the head and mane of the knight. For

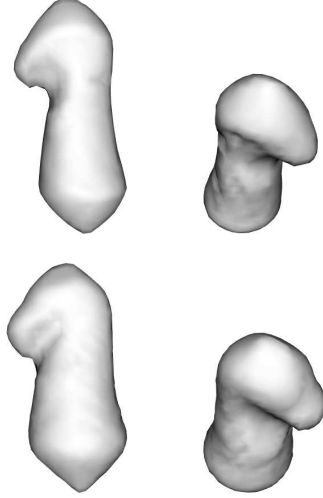


Figure 4.9: Two viewpoints of the thin lens-based (top) and pinhole-based reconstructions (bottom) for the knight dataset. The thin lens method correctly reconstructs the shape of the knight and is able to better carve out the slot between the muzzle and the body compared to the pinhole method.

the defocused images, the camera had $f = 45$ mm, $d = f/5.3$, and $z = 371$ mm. The initial surface was an ellipsoid containing the knight.

The obtained reconstructions in Figure 4.9 show that the thin lens method was able to more accurately reconstruct the knight’s shape. Without any decrease in smoothness (even though there are no explicit smoothness regularizers) the thin lens reconstruction provides increased accuracy compared to the pinhole reconstruction. For instance, the pinhole method provides an overly smooth reconstruction in order to be photometrically consistent with the defocused images, but this means that it is not able to fully carve out the niche between the knight’s muzzle and body since the niche appears more shallow (and

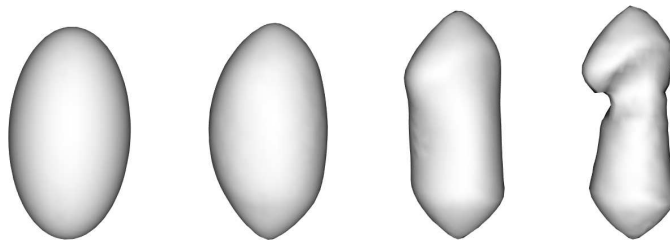


Figure 4.10: Four snapshots of the evolving surface for the chesspiece dataset using the thin lens method after 0, 20, 40, and 70 iterations

even non-existent in some images) due to defocus blur. In contrast, the full depth of this niche, as well as a flatter bottom (which was not visible in any of the images) is clearly seen in the thin lens reconstruction.

4.2 Quantitative Analysis

Quantitative results for these experiments are summarized in Table 4.1. Due to the increased number of pixels needed to process for every updated surface point, from the line integral in (3.10), the thin lens method is significantly more computationally expensive. The above experiments had iteration times ranging from 1 to 3.5 minutes using the thin lens method, with iteration time increasing with the level of defocus and number of foreground pixels. In contrast, the pinhole method ran consistently at 100 ms or less per iteration. However, the thin lens method converged faster (in terms of the number of iterations) than the pinhole method for all except the dumbbell datasets. To measure the quantitative error of the final reconstructions relative to the input images, we used the total squared-error between the predicted (calculated using the thin lens forward model) and the actual pixel intensity. For comparison purposes, error for the noisy dumbbell reconstructions were computed with respect to the noiseless images. The thin lens method gave a lower error for all datasets with the most significant difference being for the dumbbell datasets where de-

Table 4.1: Performance Comparison of Thin Lens and Pinhole Reconstruction Methods

Dataset	Error		# Iterations		Max. Iter. Time (s)	
	Thin Lens	Pinhole	Thin Lens	Pinhole	Thin Lens	Pinhole
Tetrahedron 1	1.78×10^9	2.47×10^9	120	129	57	0.07
Tetrahedron 2	9.10×10^9	1.03×10^{10}	120	160	65	0.05
Dumbbell	1.91×10^9	7.11×10^9	400	200	217	0.1
Noisy Dumbbell	1.87×10^9	7.18×10^9	480	400	217	0.1
Chess Piece	4.99×10^{10}	5.40×10^{10}	70	100	87	0.1

The thin lens reconstructions were more photometrically consistent and less affected by by noise than the pinhole reconstructions, often requiring significantly fewer iterations to converge even though each iteration itself was more costly.

focus blur had the most effect. Also, the thin lens method's robustness to noise can now be seen quantitatively as the pinhole method was affected significantly more by adding noise to the dumbbell images compared to the thin lens method, both in terms of the forward model error as well as convergence time; for instance, the thin lens method only required 80 additional iterations compared to the pinhole method's 200.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This thesis developed a novel variational algorithm for multiview reconstruction of dense surfaces that is effective when the input images are defocused as well as when the camera focal parameters change between images along with the viewpoint. It is reminiscent of depth-from-defocus as the camera is modeled as a thin lens instead of a pinhole; this allows for the successful reconstruction of sharp edges and corners that appear rounded in images due to defocus blur, as a thin lens appropriately models defocus blur as a property of the imaging process and not of the scene. The more defocus blur inherent in the images, the better this method performs over traditional pinhole-based methods due to the more general forward model developed in Chapter 2; this forward model was shown to be more accurate than existing irradiance models, especially for large lens apertures and near-focused points where the pinhole model is invalid. However, this performance does come at the cost of significantly increased computation time with the need to integrate over a sizable number of pixels for every updated surface point each iteration. Even so, this very same integration grants this thin lens-based method a form of natural regularization that increases its robustness to noise and decreases the need for artificial regularizers, and this benefit is proportional to the level of defocus. As such, the proposed thin lens-based method is an effective complement to existing methods that can be applied to a large number of previously unsupported situations where both the viewpoint and focal properties of the camera can arbitrarily change between images.

There are still open areas of research and future work that can be done to improve and better understand the proposed method. While the mathematics for both the piecewise-smooth and piecewise-constant radiance cases have been developed, only the piecewise-constant radiance version was implemented for this thesis. It still remains to implement

and validate the more general piecewise-smooth radiance version, which should require fewer images, show improved performance, and be applicable to a larger class of images. For instance, the smooth radiance algorithm should prove more effective on the chess piece images since they still contained a significant amount of irradiance variance within the foreground and background regions respectively. The more general method would also provide the means to accurately do texture mapping by mapping the optimal radiance functions onto the resulting reconstructed shapes. In the same vein, the current state of the algorithm assumes Lambertian reflectance, and work can be done to modify the algorithm for non-Lambertian scenes, which is important for reconstruction in outdoor environments that are often non-Lambertian. Work can also be done to widen the applicability to more dynamic environments as well as longer data capture times by incorporating time-varying radiance functions. Further extensions of the algorithm include a multi-scale implementation, where the usual Gaussian blurring done at each resolution scale is replaced with blurring consistent with the focal properties of the camera, as well as modeling the thin lens as an array of pinholes, which has the potential to reduce computation time without significance performance loss.

Appendices

APPENDIX A

SHAPE SENSITIVITY OF FLUX THROUGH SHAPES WITH FREE BOUNDARIES

To compute the sensitivity of the conjugate irradiance function $E(X)$ to perturbations in the surface shape, we need to compute the shape sensitivities of a class of flux integrals through an evolving subset of an evolving shape. Not only is the shape evolving, but the subset of the parameter space corresponding to the evolving portion of the shape is itself evolving.

Let $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ denote a differentiable vector field in space, (u, v) denote time-independent parameters for the evolving surface patch $S(u, v, t)$, and $U(t) \subset \mathbb{R}^2$ denote an evolving compact subset of its fixed parameter space with smooth boundary. We may now express the time-varying flux of f through the corresponding subpatch of S as

$$\text{Flux} = \int_{S(U)} f \cdot N \, dS = \int_{U(t)} f \cdot (S_u \times S_v) \, du \, dv$$

where N and dS denote the outward unit normal to S and the surface area element, respectively.

A.1 Boundary integral

The boundary ∂U of the evolving parameter subset U is represented by the parameterized planar curve

$$U^*(p, t) = (u^*(p, t), v^*(p, t))$$

where $p \in [0, 1]$ is the time-independent curve parameter. Notice that p also gives a time-independent parameterization of the evolving surface subpatch boundary S^* as

$$S^*(p, t) = S(u^*(p, t), v^*(p, t), t)$$

The total time derivative of the flux through the U -subpatch is

$$\frac{d}{dt}\text{Flux} = \int_0^1 f \cdot (S_u \times S_v)(U_t^* \cdot n^*) \|U_p^*\| dp + \int_U f_t \cdot (S_u \times S_v) + f \cdot (S_u \times S_v)_t du dv$$

where

$$n^* = Jt^* \quad \text{and} \quad t^* = \frac{U_p^*}{\|U_p^*\|} = \frac{(u_p^*, v_p^*)}{\sqrt{(u_p^*)^2 + (v_p^*)^2}}$$

respectively denote the unit normal and unit tangent to U^* . After expanding and swapping the derivative order $S_{ut} \rightarrow S_{tu}$ and $S_{vt} \rightarrow S_{tv}$ in the last term we obtain

$$\begin{aligned} \frac{d}{dt}\text{Flux} &= \int_0^1 f \cdot (S_u \times S_v)(U_t^* \cdot n^*) \|U_p^*\| dp \\ &\quad + \int_U f_t \cdot (S_u \times S_v) + f \cdot (S_{tu} \times S_v) + f \cdot (S_u \times S_{tv}) du dv \\ &= \int_0^1 f \cdot (S_u \times S_v)(U_t^* \cdot n^*) \|U_p^*\| dp \\ &\quad + \int_U f_t \cdot (S_u \times S_v) + S_{tu} \cdot (S_v \times f) + S_{tv} \cdot (f \times S_u) du dv \\ &= \int_0^1 (f \cdot (S_u \times S_v)(U_t^* \cdot n^*) + S_t \cdot (S_v \times f)n_1^* + S_t \cdot (f \times S_u)n_2^*) \|U_p^*\| dp \\ &\quad + \int_U f_t \cdot (S_u \times S_v) - S_t \cdot ((S_v \times f)_u + (f \times S_u)_v) du dv \\ &= \int_0^1 (f \cdot (S_u \times S_v)(U_t^* \cdot n^*) + S_t \cdot (S_u t_1^* + S_v t_2^*)) \|U_p^*\| dp \\ &\quad + \int_U f_t \cdot (S_u \times S_v) - S_t \cdot (S_v \times f_u + f_v \times S_u) du dv \end{aligned}$$

where n_i^* and t_i^* denote the i -th component of n^* and t^* . Finally, substituting the relations

$$\begin{aligned}
S_p^* &= S_u u_p + S_v v_p \\
\frac{dS^*}{dt} &= S_u u_t^* + S_v v_t^* + S_t \\
\frac{dS^*}{dt} \times S_p^* &= (S_u \times S_v)(u_t^* v_p - v_t^* u_p) + S_t \times (S_u u_p + S_v v_p) \\
&= (S_u \times S_v)(U_t^* \cdot n^*) \|U_p^*\| + S_t \times (S_u t_1^* + S_v t_2^*) \|U_p^*\|
\end{aligned}$$

we obtain

$$\frac{d}{dt} \text{Flux} = \int_0^1 f \cdot \left(\frac{dS^*}{dt} \times S_p^* \right) dp + \int_U f_t \cdot (S_u \times S_v) - S_t \cdot (S_v \times f_u + f_v \times S_u) du dv$$

A.2 Surface integral

We now simplify the remaining surface integral, letting Df denote the Jacobian of the vector field f :

$$\begin{aligned}
&\int_U f_t \cdot (S_u \times S_v) - S_t \cdot (S_v \times f_u + f_v \times S_u) du dv \\
&= \int_U (S_u \times S_v) \cdot (Df) S_t + S_t \cdot \left(((Df) S_u) \times S_v + S_u \times ((Df) S_v) \right) du dv \\
&= \int_U (S_u \times S_v) \cdot (Df) S_t \\
&\quad + S_t \cdot \left(((Df) S_u - S_u) \times (S_v - (Df) S_v) + ((Df) S_u) \times ((Df) S_v) + S_u \times S_v \right) du dv \\
&= \int_U (S_u \times S_v) \cdot (Df) S_t \\
&\quad + S_t \cdot \left(-\text{adj}^T(\mathcal{I} - Df)(S_u \times S_v) + \text{adj}^T(Df)(S_u \times S_v) + S_u \times S_v \right) du dv \\
&= \int_U (S_u \times S_v) \cdot (Df) S_t + S_t \cdot \left(\text{adj}^T(Df) - \text{adj}^T(\mathcal{I} - Df) + \mathcal{I} \right) (S_u \times S_v) du dv
\end{aligned}$$

where $\text{adj}(A)$ denotes the adjugate matrix for a given matrix A and \mathcal{I} is the identity matrix.

We now employ the following formula for any 3x3 matrix A :

$$\text{adj}(A) = \frac{(\text{trace } A)^2 - \text{trace } A^2}{2} \mathcal{I} - (\text{trace } A)A + A^2$$

to obtain

$$\text{adj}(A) - \text{adj}(\mathcal{I} - A) + \mathcal{I} = \mathcal{I} \text{trace } A - A$$

and therefore continue to simplify the surface integral:

$$\begin{aligned} &= \int_U S_t \cdot \left((Df)^T (S_u \times S_v) \right) + S_t \cdot \left(\mathcal{I} \text{trace}(Df) - (Df)^T \right) (S_u \times S_v) du dv \\ &= \int_U \text{trace}(Df) S_t \cdot (S_u \times S_v) du dv \\ &= \int_{S(U)} (\nabla \cdot f) S_t \cdot N dS \end{aligned}$$

A.3 Combined result

Finally, we express the combined result for the total time derivative of flux as

$$\frac{d}{dt} \text{Flux} = \int_{S^*} f \cdot \left(\frac{dS^*}{dt} \times T^* \right) ds^* + \int_{S(U)} \nabla \cdot f (S_t \cdot N) dS \quad (\text{A.1})$$

where T^* denotes the unit tangent vector along the surface subpatch boundary S^* (which will also be a tangent vector to the surface S) and ds^* denotes its arclength element. Note that while formulas for flux derivatives have been previously given in [103, 104], these formulas only consider the case of a moving surface, not a deforming surface. The main consequence of this is that the partial time derivative in those formulations is replaced with the material derivative as given by the $\frac{dS^*}{dt}$ term in (A.1).

APPENDIX B

SWAPPING ORDER OF INTEGRATION FOR OCCLUDING BOUNDARY INTEGRAL

B.1 Occluding boundary and occluding projection

We may express the occluding surface boundary $S_{(\Gamma_S(X))}^*$ (with respect to a fixed viewpoint $P(X)$ on the focal plane) as the zero level set of the following function $g : S \rightarrow \mathbb{R}$ on the surface:

$$g(S, X) \doteq (S - P(X)) \cdot N \tag{B.1}$$

Similarly, if we fix a point S on the surface, the zero level set of (B.1) on the focal plane domain expresses the occluding projection contour $\Omega_{(\Gamma_S(S))}^*$ which is the locus of points in Ω for which S is part of the occluding boundary for $P(X)$.

It will be necessary to use the gradients of g with respect to both X and S . The gradient and gradient magnitude with respect to X is

$$\begin{aligned} \nabla_X g &= N - (N \cdot e_z) e_z \\ \|\nabla_X g\| &= \sqrt{1 - (N \cdot e_z)^2} \end{aligned} \tag{B.2}$$

Using the Weingarten equations [105] and the radial/transverse frame (3.3), the (intrinsic) gradient and gradient magnitude with respect to S is found to be

$$\begin{aligned}\nabla_S g &= -r \mathbb{S}(T_r) = r(\kappa_r T_r - \tau_r T_\perp) \\ \|\nabla_S g\| &= r \|T_r\| \sqrt{\kappa_r^2 + \tau_r^2}\end{aligned}\tag{B.3}$$

where $\mathbb{S}(\cdot)$ is the shape operator of S , and κ_r and τ_r are the geodesic curvature and torsion of S in the direction of T_r .

B.2 Dirac integration formula

Combining the generalized scaling property of the Dirac delta with the coarea formula gives the following integration formula (assuming a differentiable scalar function $g : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\|\nabla g \neq \mathbf{0}\|$):

$$\begin{aligned}\int_{\mathbb{R}^n} f(\mathbf{x}) \delta(g(\mathbf{x})) d\mathbf{x} &= \int_{g^{-1}(0)} \frac{f(u, v)}{\|\nabla g\|} d\mathcal{H}_{n-1}(\mathbf{x}) \\ \int_{g^{-1}(0)} f d\mathcal{H}_{n-1} &= \int_{\mathbb{R}^n} f \|\nabla g\| \delta(g(\mathbf{x})) d\mathbf{x}\end{aligned}\tag{B.4}$$

where \mathcal{H}_{n-1} denotes the $n - 1$ dimensional Hausdorff measure of the zero level set of g [106].

B.3 Planar integration of occluding boundary to surface integration of occluding projection

We now consider an integral over all reference points $P(X)$ in the focal plane of occluding boundary integrals with respect to each surface point. We begin by applying (B.4) with g defined as in (B.1) and having gradient magnitude (B.3). After swapping the order of integration, the Dirac integration formula is applied again, but in the reverse direction, using (B.2):

$$\begin{aligned}
& \int_{\Omega} \left(\int_{S_{(\Gamma_S(X))}^*} f \, ds^* \right) dX \\
&= \int_{\Omega} \int_{S_{(\Gamma_S(X))}} fr \|T_r\| \sqrt{\kappa_r^2 + \tau_r^2} \delta((S - P) \cdot N) \, dS dX \\
&= \int_S \int_{\Omega_{(\Gamma_S(S))}} fr \|T_r\| \sqrt{\kappa_r^2 + \tau_r^2} \delta((S - P) \cdot N) \, dX dS \\
&= \int_S \int_{\Omega_{(\Gamma_S(S))}} fr \|T_r\| \sqrt{\kappa_r^2 + \tau_r^2} \frac{\|\nabla_X g\|}{\|\nabla_X g\|} \delta((S - P) \cdot N) \, dX dS \\
&= \int_S \int_{\Omega_{(\Gamma_S(S))}^*} fr^* \sqrt{\frac{(\kappa_r^*)^2 + (\tau_r^*)^2}{1 - (N^* \cdot e_z)^2}} \, ds_X dS
\end{aligned} \tag{B.5}$$

APPENDIX C

PRELIMINARY COMPUTATIONS (2D)

C.1 Full Jacobian for Change of Variables (2D)

In this 2D case, the surface is simply a curve $C(s)$, where s is the arclength parameter; likewise the focal plane is simply a line. Consider a ray emitted from $C(s)$ that passes through point $P(x)$ on the focal plane; for sake of space and neatness these points will simply be denoted henceforth as C and P . Let r denote the length of the ray segment connecting C and P and $e_r = (C - P)/r$ denote the unit ray vector. Let (T, N) and (e_x, e_z) denote the unit tangents and normals to C and P , respectively. Finally, as previously defined, θ and α are the angles that e_r makes with N and e_z , respectively. Table C.1 summarizes the inner product relationships between all these unit vectors in terms of the ray angles θ and α . Note that only $N \cdot e_r$ and $e_z \cdot e_r$ remain unchanged in 3D case

Table C.1: 2D Angle Relationships

$e_x \cdot e_r = -\sin \alpha$	$e_z \cdot e_r = \cos \alpha$
$T \cdot e_r = \sin \theta$	$N \cdot e_r = -\cos \theta$
$T \cdot e_x = -\cos(\theta - \alpha)$	$N \cdot e_x = -\sin(\theta - \alpha)$
$T \cdot e_z = \sin(\theta - \alpha)$	$N \cdot e_z = -\cos(\theta - \alpha)$

C.1.1 Jacobian $\frac{\partial(s, \theta)}{\partial(x, \alpha)}$ of surface coordinates $s(x, \alpha)$ and $\theta(x, \alpha)$

We first differentiate in α starting with

$$\frac{\partial e_r}{\partial \alpha} = \frac{I - e_r e_r^T}{r} \frac{\partial(C - P)}{\partial \alpha} = \frac{I - e_r e_r^T}{r} T \frac{\partial s}{\partial \alpha} = \frac{T - e_r \sin \theta}{r} \frac{\partial s}{\partial \alpha}$$

Now differentiating $\cos \alpha = e_r \cdot e_z$ yields

$$-\sin \alpha = \frac{\partial e_r}{\partial \alpha} \cdot e_z = \frac{-\cos \theta \sin \alpha}{r} \frac{\partial s}{\partial \alpha} \Rightarrow \frac{\partial s}{\partial \alpha} = \frac{r}{\cos \theta} \quad (\text{C.1})$$

Next, differentiating $-\cos \theta = e_r \cdot N$ yields

$$\sin \theta \frac{\partial \theta}{\partial \alpha} = \frac{\partial e_r}{\partial \alpha} \cdot N + e_r \cdot \frac{\partial N}{\partial \alpha} = \sin \theta \left(\frac{\cos \theta}{r} + \kappa \right) \frac{\partial s}{\partial \alpha}$$

where κ denotes the curvature of C . Substitution by (C.1) then gives

$$\frac{\partial \theta}{\partial \alpha} = 1 + \frac{\kappa r}{\cos \theta} \quad (\text{C.2})$$

Next, we differentiate in x starting with

$$\begin{aligned} \frac{\partial e_r}{\partial x} &= \frac{I - e_r e_r^T}{r} \frac{\partial (C - P)}{\partial x} = \frac{I - e_r e_r^T}{r} \left(T \frac{\partial s}{\partial x} - e_x \right) \\ &= \frac{(T - e_r \sin \theta) \frac{\partial s}{\partial x} - e_x - e_r \sin \alpha}{r} \end{aligned}$$

Now differentiating $\cos \alpha = e_r \cdot e_z$ yields

$$0 = \frac{(-\cos \theta \sin \alpha) \frac{\partial s}{\partial x} - \cos \alpha \sin \alpha}{r} \Rightarrow \frac{\partial s}{\partial x} = -\frac{\cos \alpha}{\cos \theta} \quad (\text{C.3})$$

Next, differentiating $-\cos \theta = e_r \cdot N$ yields

$$\sin \theta \frac{\partial \theta}{\partial x} = \frac{\partial e_r}{\partial x} \cdot N + e_r \cdot \frac{\partial N}{\partial x} = \sin \theta \left(\frac{\cos \theta \frac{\partial s}{\partial x} + \cos \alpha}{r} + \kappa \frac{\partial s}{\partial x} \right)$$

Substitution by (C.3) then gives the last entry of the Jacobian

$$\sin \theta \frac{\partial \theta}{\partial x} = \sin \theta \left(-\kappa \frac{\cos \alpha}{\cos \theta} \right) \Rightarrow \frac{\partial \theta}{\partial x} = -\kappa \frac{\cos \alpha}{\cos \theta} \quad (\text{C.4})$$

Combining (C.3), (C.1), (C.4), and (C.2) into matrix form gives the following Jacobian and

determinant

$$\frac{\partial(s, \theta)}{\partial(x, \alpha)} = \begin{bmatrix} \frac{\partial s}{\partial x} & \frac{\partial s}{\partial \alpha} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial \alpha} \end{bmatrix} = \frac{1}{\cos \theta} \begin{bmatrix} -\cos \alpha & r \\ -\kappa \cos \alpha & \cos \theta + \kappa r \end{bmatrix} \quad (\text{C.5a})$$

$$\left| \frac{\partial(s, \theta)}{\partial(x, \alpha)} \right| = \left| -\frac{\cos \alpha}{\cos \theta} \right| = \frac{\cos \alpha}{\cos \theta} \quad (\text{C.5b})$$

Note that the absolute value can be removed since $\cos \alpha$ and $\cos \theta$ will always have the same sign.

C.1.2 Jacobian $\frac{\partial(x, \alpha)}{\partial(s, \theta)}$ of surface coordinates $x(s, \theta)$ and $\alpha(s, \theta)$

We first differentiate in θ starting with

$$\frac{\partial e_r}{\partial \theta} = \frac{I - e_r e_r^T}{r} \frac{\partial(C - P)}{\partial \theta} = \frac{I - e_r e_r^T}{r} \left(-e_x \frac{\partial x}{\partial \theta} \right) = \frac{-e_x - e_r \sin \alpha}{r} \frac{\partial x}{\partial \theta}$$

Now differentiating $-\cos \theta = e_r \cdot N$ yields

$$-\sin \theta = \frac{\partial e_r}{\partial \theta} \cdot N = \frac{\sin \theta \cos \alpha}{r} \frac{\partial x}{\partial \theta} \Rightarrow \frac{\partial x}{\partial \theta} = \frac{r}{\cos \alpha} \quad (\text{C.6})$$

whereas differentiating $-\cos \alpha = e_r \cdot e_z$ yields

$$-\sin \alpha \frac{\partial \alpha}{\partial \theta} = \frac{-\cos \alpha \sin \alpha}{r} \frac{\partial x}{\partial \theta}$$

Substitution by (C.6) gives us

$$\frac{\partial \alpha}{\partial \theta} = 1 \quad (\text{C.7})$$

Next, we differentiate in s starting with

$$\begin{aligned}\frac{\partial e_r}{\partial s} &= \frac{I - e_r e_r^T}{r} \frac{\partial(C - P)}{\partial s} = \frac{I - e_r e_r^T}{r} \left(T - e_x \frac{\partial x}{\partial s} \right) \\ &= \frac{(T - e_r \sin \theta) - (e_x + e_r \sin \alpha) \frac{\partial x}{\partial s}}{r}\end{aligned}$$

Now differentiating $-\cos \theta = e_r \cdot N$ yields

$$0 = \sin \theta \left(\frac{\cos \theta + \cos \alpha \frac{\partial x}{\partial s}}{r} + \kappa \right) \Rightarrow \frac{\partial x}{\partial s} = -\frac{\cos \theta + \kappa r}{\cos \alpha} \quad (\text{C.8})$$

whereas differentiating $\cos \alpha = e_r \cdot e_z$ yields

$$-\sin \alpha \frac{\partial \alpha}{\partial s} = -\sin \alpha \frac{\cos \theta + \cos \alpha \frac{\partial x}{\partial s}}{r}$$

Substitution by (C.8) then gives us

$$\frac{\partial \alpha}{\partial s} = -\kappa \quad (\text{C.9})$$

Combining (C.8), (C.6), (C.9), and (C.7) into matrix form gives the following Jacobian and determinant

$$\frac{\partial(x, \alpha)}{\partial(s, \theta)} = \begin{bmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial \theta} \\ \frac{\partial \alpha}{\partial s} & \frac{\partial \alpha}{\partial \theta} \end{bmatrix} = \frac{1}{\cos \alpha} \begin{bmatrix} -\cos \theta - \kappa r & r \\ -\kappa \cos \alpha & \cos \alpha \end{bmatrix} \quad (\text{C.10a})$$

$$\left| \frac{\partial(x, \alpha)}{\partial(s, \theta)} \right| = \left| -\frac{\cos \theta}{\cos \alpha} \right| = \frac{\cos \theta}{\cos \alpha} \quad (\text{C.10b})$$

Note that the absolute value can be removed since $\cos \alpha$ and $\cos \theta$ will always have the same sign.

C.2 Image Irradiance Equation (2D)

In the much simpler 2D case, each ray has only one angular parameter and one spatial parameter, either (s, θ) in surface coordinates or (x, α) in focal plane coordinates. It is easy to see that the extreme ray angles α_{\pm} are related to the spatial coordinates by

$$\tan \alpha_{\pm}(x) = -\frac{x}{z} \pm \frac{d}{2z}$$

Now, the 2D conjugate irradiance equation (assuming an in-focus point and Lambertian reflectance) is given by

$$\begin{aligned} E(x) &= L(s) \int_{\alpha_{-}(x)}^{\alpha_{+}(x)} \cos \alpha \, d\alpha \\ &= L(s) (\sin \alpha_{+} - \sin \alpha_{-}) \\ &= L(s) \left(\sin \tan^{-1} \left(\frac{-x + d/2}{z} \right) - \sin \tan^{-1} \left(\frac{-x - d/2}{z} \right) \right) \\ &= L(s) \left(\frac{x + d/2}{\sqrt{(x + d/2)^2 + z^2}} - \frac{x - d/2}{\sqrt{(x - d/2)^2 + z^2}} \right) \end{aligned} \quad (\text{C.11})$$

Next, in 2D the relationship between the conjugate irradiance and the image irradiance is similar in form to its 3D counterpart; the main difference is that there is only one factor of the lens magnification instead of two:

$$E'(x') = \frac{1}{M} E(x) \quad (\text{C.12})$$

where $M = 1 - z/f$ and $x' = x/M$ as before. Now combining (C.11) and (C.12) gives the final closed-form of the 2D image irradiance:

$$E'(x') = \frac{1}{M} L(s) \left(\frac{x + d/2}{\sqrt{(x + d/2)^2 + z^2}} - \frac{x - d/2}{\sqrt{(x - d/2)^2 + z^2}} \right) \quad (\text{C.13})$$

C.3 Shape Sensitivity of Flux (2D)

Similar to A, let $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a differentiable vector field in the plane, and let C denote an evolving curve. Let p be a time-independent parameter for $C(p, t)$, and let $U(t) \subset \mathbb{R}$ represent an evolving set of $n \geq 1$ intervals $\{[p_1, p_2], [p_3, p_4], \dots, [p_{n-1}, p_n]\}$ within its fixed parameter space. The time-varying flux through the corresponding subset of C can be expressed as

$$\text{Flux} = \int_{C(U)} f \cdot N \, ds = \int_{U(t)} f \cdot JC_p \, dp$$

where N denotes the unit normal to C and ds its arclength element, and $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$.

C.3.1 Boundary terms

The boundary $\partial U = U^*(t)$ of the evolving parameter subset U can be represented by the pairs of sub-interval endpoints

$$U^*(t) = \left\{ \underbrace{p_1(t), p_2(t)}_{\text{sub-interval 1}}, \underbrace{p_3(t), p_4(t)}_{\text{sub-interval 2}}, \dots, \underbrace{p_{n-1}(t), p_n(t)}_{\text{sub-interval } n} \right\}$$

In the following, p^* refers arbitrarily to any of these sub-interval endpoints. Accordingly, the corresponding contour subsection endpoint C^* can be represented as

$$C^*(t) = C(p^*(t), t)$$

in order to denote the actual points on the curve at integration boundaries. The total U -subsection flux derivative can then be written as

$$\begin{aligned}
\frac{d}{dt}\text{Flux} &= \sum_{k=1}^{2n} f \cdot JC_p \frac{dp_k}{dt} (-1)^k + \int_U f_t \cdot JC_p + f \cdot JC_{pt} dp \\
&= \sum_{k=1}^{2n} f \cdot J \left(C_p \frac{dp_k}{dt} + C_t \right) (-1)^k + \int_U f_t \cdot JC_p - f_p \cdot JC_t dp \\
&= \sum_{k=1}^{2n} f \cdot J \frac{C^*}{dt} (-1)^k + \int_U f_t \cdot JC_p - f_p \cdot JC_t dp \tag{C.14}
\end{aligned}$$

where the second line is obtained by swapping the derivative order $C_{pt} \rightarrow C_{tp}$ and applying integration by parts, and the final line is obtained by expanding the material derivative

$$\frac{dC^*}{dt} = C_p \frac{dp^*}{dt} + C_t$$

C.3.2 Contour Integral

The remaining contour integral can be simplified as follows, with Df denoting the Jacobian of the vector field f :

$$\begin{aligned}
\int_U f_t \cdot JC_p - f_p \cdot JC_t dp &= \int_U JC_p \cdot (Df)C_t - JC_t \cdot (Df)C_p dp \\
&= \int_{C(U)} N \cdot (Df)C_t - JC_t \cdot (Df)T ds \\
&= \int_{C(U)} N^T (Df) ((C_t \cdot T)T + (C_t \cdot N)N) \\
&\quad - ((C_t \cdot T)N - (C_t \cdot N)T)^T (Df)T ds \\
&= \int_{C(U)} (T^T (Df)T + N^T (Df)N) (C_t \cdot N) ds \\
&= \int_{C(U)} \nabla \cdot f (C_t \cdot N) ds \tag{C.15}
\end{aligned}$$

C.3.3 Combined result

Substituting (C.15) into (C.14) gives the final result

$$\frac{d}{dt}\text{Flux} = \sum_{k=1}^{2n} f \cdot J \frac{dC^*}{dt} (-1)^k + \int_{C(U)} \nabla \cdot f(C_t \cdot N) ds \quad (\text{C.16})$$

Note that the contour integral is exactly the 2D analog of the surface integral in the 3D case (A.1). Also, the first terms are functionally identical, as they are respectively summing and integrating along the boundary an inner product between the vector field and vectors orthogonal to the boundary points. However, this set of boundary points can be discretized in the 2D case while they span a continuum in the 3D case.

C.4 Surface Evolution Equation (2D)

Irradiance sensitivity

Just as in the 3D case, the irradiance $E(x)$ can be written as a flux integral:

$$E(x) = - \int_{C(\Gamma_C(x))} \frac{\hat{Q} \cos \alpha}{r} e_r \cdot N ds \quad (\text{C.17})$$

Note that the only main difference here in 2D is that there is only one factor of r in the denominator instead of two, which is to be expected. Now, applying (C.16) gives the total time derivative of E as

$$\frac{dE}{dt}(x) = \sum_{k=1}^{2n} \left(-\frac{\hat{Q} \cos \alpha^*}{r^*} e_r^* \right) \cdot J \frac{dC^*}{dt} (-1)^k - \int_{C(\Gamma_C(x))} \nabla \hat{Q} \cdot e_r \frac{\cos \alpha}{r} (C_t \cdot N) ds$$

Interior boundary points (non-occluding) A point $C^* \in \partial C_{\Gamma_C(x)}$ which does not fall along a visibility boundary with respect to $P(x)$ represents the endpoint of a ray along the boundary $\partial \Gamma_C(x)$ of the ray set between $P(x)$ and the lens, emitted from C . For such

points, the implications are the same as in 3D: the ray direction e_r^* at the boundary remains unchanged even as C^* evolves (e.g. C^* moves along the ray):

$$e_r^* = \frac{C^* - P}{r^*} = \text{constant} \quad \xrightarrow{\frac{d}{dt}} \quad \frac{dC^*}{dt} = \frac{dr^*}{dt} e_r^*$$

As in the 3D case, such points do not contribute to the boundary summation since $e_r^* \cdot J e_r^* = 0$, and may therefore be ignored.

Occluding boundary points As such, attention may again be restricted to the subset of $\partial C_{(\Gamma_C(x))}$ representing the occluding boundaries with respect to $P(x)$, which are the points C^* whose rays to $P(x)$ satisfy the occluding boundary condition

$$e_r^* \cdot N^* = 0$$

By decomposing C_t into its tangential and normal components, it can be found that the inner product in the boundary summation can be written as

$$e_r^* \cdot J \frac{dC^*}{dt} = - \left(\underbrace{C_t(s^*, t)}_{\doteq C_t^*} \cdot N^* \right) (e_r^* \cdot T^*)$$

Finally, noting that $e_r^* \cdot T^* = (-1)^k$, the time derivative of E simplifies to

$$\frac{dE}{dt}(x) = \sum_{\substack{\text{occluding} \\ C_{(\Gamma_C(x))}}} \left(\frac{\hat{Q} \cos \alpha^*}{r^*} \right) (C_t^* \cdot N^*) - \int_{C_{(\Gamma_C(x))}} \nabla \hat{Q} \cdot e_r \frac{\cos \alpha}{r} (C_t \cdot N) ds \quad (\text{C.18})$$

Total matching error sensitivity

The total matching error (for a single image), and its time derivative, is simply the 2D analogues of their respective equations from the 3D case:

$$J = \frac{M}{2} \int_{\Omega} \mathcal{E}^2 dx \quad (\text{C.19a})$$

$$\frac{dJ}{dt} = \int_{\Omega} \mathcal{E} \frac{dE}{dt} dx \quad (\text{C.19b})$$

Boundary term Inserting the boundary term of (C.18) into (C.19b) and again utilizing the Dirac integration formula (B.4) gives

$$\begin{aligned} & \int_{\Omega} \mathcal{E}(x) \left(\sum_{\substack{\text{occluding} \\ C_{(\Gamma_C(x))}}} \frac{\hat{Q}^* \cos \alpha^*}{r^*} (C_t^* \cdot N^*) \right) dx \\ &= \int_{\Omega} \mathcal{E}(x) \left(\int_C L \cos \alpha (C_t \cdot N) |\kappa e_r \cdot T| \delta((C - P) \cdot N) ds \right) dx \\ &= \int_C L |\kappa| \left(\int_{P_{(\Gamma_C(C))}} \mathcal{E}(x) \cos \alpha |e_r \cdot T| \underbrace{\delta((C - P) \cdot N)}_{\substack{0 \text{ or } 1 \text{ root} \\ \text{(at } \theta = \pm \frac{\pi}{2})}} dx \right) (C_t \cdot N) ds \\ &= \int_C L |\kappa| \left[\mathcal{E}(x) \cos \alpha \frac{\overbrace{|e_r \cdot T|}^1}{\underbrace{|-e_x \cdot N|}_{\cos \alpha}} \right]_{\theta = \pm \frac{\pi}{2}} (C_t \cdot N) ds \\ &= \int_C L |\kappa| [\mathcal{E}(x)]_{\theta = \pm \frac{\pi}{2}} (C_t \cdot N) ds \end{aligned}$$

Both terms Noting that the curvature κ must be positive at an occluding boundary point allows the absolute value sign to be dropped, and the final simplified expressions for the matching error sensitivity and the resulting evolution flow for C for a single image are

given as

$$\frac{dJ}{dt} = \int_{C^*} Q\mathcal{E}(x^*)\kappa^*(C_t^* \cdot N) ds^* - \int_C \left(\nabla \hat{Q} \cdot \int_{P(\Gamma_C(C))} \mathcal{E}(x)e_r \frac{\cos \alpha}{r} dx \right) (C_t \cdot N) ds \quad (\text{C.20})$$

$$\begin{aligned} C_t(s) &= \left(-Q\mathcal{E}(x^*)\kappa^* + \nabla \hat{Q} \cdot \int_{x^-}^{x^+} \mathcal{E}(x)e_r \frac{\cos \alpha}{r} dx \right) N \\ &= \left(-Q\mathcal{E}(x^*)\kappa^* + \nabla \hat{Q} \cdot \int_{\theta^-}^{\theta^+} \mathcal{E}(x)e_r d\theta \right) N \end{aligned} \quad (\text{C.21})$$

It can be seen that (C.21) is similar in form to its 3D analog (3.10). For one thing, the second (non-boundary) term is identical in both (with the accepted idea that angle and solid angle elements are corresponding analogues). The first (boundary) terms differ in three important ways. First is that in 2D for a given occluding boundary point on the curve, there is only a single point in the focal plane that forms an occluding boundary ray with that curve point; however, in 3D there is an entire line of such focal plane points. Second, in both 2D and 3D one factor of r^{-1} is canceled out from applying the Dirac integration formula, but this still leaves one factor in the 3D case due to the increase in dimensionality. Finally, the factor of $\cos \alpha$ is not completely canceled out in the 3D case. Expanding out

$$\sqrt{1 - (N \cdot e_z)^2} = \sqrt{\cos^2 \alpha + ((e_z \times e_r) \cdot N)^2}$$

it is seen that only when N lies in the plane spanned by e_r and e_z does the above simplify to $\cos \alpha$. This is the only possible scenario in 2D, but it is far from guaranteed in 3D, which explains both the new radical term and the non-cancellation of the cosine term in the 3D case.

REFERENCES

- [1] C. Rothwell, O. Faugeras, and G. Csurka, “A comparison of projective reconstruction methods for pairs of views,” *Computer Vision and Understanding*, vol. 68, no. 1, pp. 37–58, Oct. 1997.
- [2] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2006, pp. 519–528.
- [3] T. Jadhav, K. Singh, and A. Abhyankar, “A review and comparison of multi-view 3d reconstruction methods,” *Journal of Engineering Research*, vol. 5, no. 3, Sep. 2017.
- [4] X. Han, H. Laga, and M. Bennamoun, “Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era,” *CoRR*, vol. abs/1906.06543, 2019. arXiv: 1906.06543.
- [5] G. Adiv, “Determining three-dimensional motion and structure from optical flow generated by several moving objects,” *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 4, pp. 348–401, 1985.
- [6] O. D. Faugeras, F. Lustman, and G. Toscani, “Motion and structure from motion from point and line matches,” in *ICCV 1987*, 1987.
- [7] B. K. P. Horn and E. J. Weldon, “Direct methods for recovering motion,” *International Journal of Computer Vision*, vol. 2, no. 1, pp. 51–76, Jun. 1988.
- [8] A. Verri, F. Girosi, and V. Torre, “Differential techniques for optical flow,” *J. Opt. Soc. Am. A.*, no. 5, pp. 912–922, 1990.
- [9] T. E. Boult and L. G. Brown, “Factorization-based segmentation of motions,” in *Proceedings of the IEEE Workshop on Visual Motion*, Oct. 1991, pp. 179–186.
- [10] D. Heeger and A. Jepson, “Subspace methods for recovering rigid motion i: Algorithm and implementation,” University of Toronto-CS Dept., Tech. Rep. RBCV TR-90-35, Nov. 1990, revised July 1991.
- [11] J. Barron, D. Fleet, and S. Beauchemin, “Performance of optical flow techniques,” Queen’s University Kingston, Ontario, Robotics and perception laboratory, Tech. Rep. RPL-TR 9107, Nov. 1992, Also in Proc. CVPR 1992, pp. 236-242.

- [12] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. of Computer Vision*, no. 2, pp. 137–154, 1992.
- [13] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," in *Computer Vision - ECCV94*, J.-O. Eklundh, Ed., Berlin: Springer-Verlag, 1994, pp. 97–108.
- [14] C. Tomasi and J. Shi, "Image deformations are better than optical flow," *Mathematical and Computer Modeling*, no. 5/6, pp. 165–175, 1996.
- [15] Y. Ma, J. Kosecka, and S. Sastry, "Motion recovery from image sequences: Discrete viewpoint vs differential viewpoint," in *Proc. Eur. Conf. Comp. Vision*, 1998.
- [16] A. Chiuso, R. Brockett, and S. Soatto, "Optimal structure from motion: Local ambiguities and global estimates," *Int. J. of Computer Vision (in press)*, 2000.
- [17] O. Faugeras, *Three dimensional vision, a geometric viewpoint*. MIT Press, 1993.
- [18] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, "An invitation to 3D vision: From images to geometric models," in Springer, 2004, ch. 3.
- [19] S. Maybank, "Theory of reconstruction from image motion," in *Information Sciences*, vol. 24, Springer-Verlag, 1992.
- [20] A. Shashua, "On geometric and algebraic aspects of 3D affine and projective structure from perspective 2D views," in *Proc. of Applications of Invariance in Computer Vision*, J. L. Mundy, A. Zisserman, and D. Forsyth, Eds., ser. Lecture Notes in Computer Science, vol. 825, Springer, 1994, pp. 127–143.
- [21] W. Triggs, "Matching constraints and the joint image," in *Proc. of the Intl. Conf. on Comp. Vision*, 1995, pp. 338–343.
- [22] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 1, pp. 13–27, Jan. 1984.
- [23] S. Uras, F. Girosi, A. Verri, and V. Torre, "A quantitative approach to motion perception," *Biol. Cybern.*, pp. 79–97, 1988.
- [24] J. Weng, T. Huang, and N. Ahuja, "Motion and structure from two perspective views: Algorithms, error analysis and error estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 5, pp. 451–476, 1989.

- [25] M. E. Spetsakis and Y. Aloimonos, "Structure from motion using line correspondences," *Int. J. of Computer Vision*, pp. 171–183, 1990.
- [26] J. Philip, "Estimation of three-dimensional motion of rigid objects from noisy observations," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 1, pp. 61–66, 1991.
- [27] M. E. Spetsakis and Y. Aloimonos, "A multiframe approach to visual motion perception," *Int. J. of Computer Vision*, pp. 245–255, 1991.
- [28] J. Weng, N. Ahuja, and T. Huang, "Motion and structure from point correspondences with error estimation: Planar surfaces," *IEEE Trans. Signal Processing*, no. 12, pp. 2691–2716, 1991.
- [29] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from line correspondences: Closed-form solution, uniqueness and optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 3, pp. 318–336, 1992.
- [30] J. I. Thomas, A. Hanson, and J. Oliensis, "Understanding noise: The critical role of motion error in scene reconstruction," in *Proc. of ICCV 93*, Berlin, 1993.
- [31] C. Taylor and D. Kriegman, "Structure and motion from line segments in multiple images," Yale University, Tech. Rep. Technical Report 9402, Jan. 1994.
- [32] Q. T. Luong and O. D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis," *Int. J. of Computer Vision*, no. 1, pp. 43–76, 1996.
- [33] A. Heyden and K. Astrom, "Euclidean reconstruction from image sequences with varying and unknown focal length and principal point," in *Proc. of the IEEE CVPR*, 1997, pp. 438–445.
- [34] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun, "Structure from motion without correspondence," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 2, Jun. 2000, 557–564 vol.2.
- [35] F. Dellaert, S. M. Seitz, S. Thrun, and C. E. Thorpe, "Feature correspondence: A markov chain monte carlo approach," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., MIT Press, 2001, pp. 852–858.
- [36] K. Ni, D. Steedly, and F. Dellaert, "Out-of-core bundle adjustment for large-scale 3D reconstruction," in *2007 IEEE 11th International Conference on Computer Vision*, Oct. 2007, pp. 1–8.
- [37] J. Bian, W. Lin, and Y. L. et. al., "Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence," *Int J Comput Vis*, pp. 1580–1593, Dec. 2020.

- [38] P. F. McLauchlan, "Gauge independence in optimization algorithms for 3D vision," in *Proc. ICCV '99 Vision Algorithms Workshop*, 1999.
- [39] E. H. Adelson, "Layered representations for vision and video," in *Proceedings IEEE Workshop on Representation of Visual Scenes (In Conjunction with ICCV'95)*, Jun. 1995, pp. 3–9.
- [40] H. S. Sawhney, "Simplifying motion and structure analysis using planar parallax and image warping," in *Proc. of the Int. Conf. on Pattern Recognition*, Seattle, Jun. 1994.
- [41] R. Kumar, P. Anandan, and K. Hanna, "Shape recovery from multiple views: A parallax-based approach," in *Proc. of the Image Understanding Workshop*, 1994.
- [42] J. Bergen, R. Kumar, P. Anandan, and M. Irani, "Representation of scenes from collections of images," in *Proc. of the IEEE Workshop on Visual Scene Representation*, Boston, Jun. 1995, pp. 10–17.
- [43] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, "Using plane + parallax for calibrating dense camera arrays," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, Jun. 2004, pp. I–I.
- [44] C. Rother and S. Carlsson, "Linear multi view reconstruction and camera recovery using a reference plane," *International Journal of Computer Vision*, vol. 49, no. 2, pp. 117–141, Sep. 2002.
- [45] K. Kutulakos and S. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.
- [46] P. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry and image-based approach," in *Proc. of the ACM SIGGRAPH*, 1996.
- [47] N. Dahiya, A. Yezzi, M. Piccinelli, and E. Garcia, "Integrated 3D anatomical model for automatic myocardial segmentation in cardiac ct imagery," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–17, Mar. 2019.
- [48] S. Osher and J. Sethian, "Fronts propogating with curvature-dependent speed: Algorithms based on hamilton-jacobi equations," *J. of Comp. Physics*, vol. 79, pp. 12–49, 1988.
- [49] S. Osher, "Level set methods," in *Geometric Level Set Methods in Imaging, Vision, and Graphics*, New York: Springer-Verlag, 2003, pp. 3–20.

- [50] O. Faugeras and R. Keriven, “Variational principles, surface evolution, pdes, level set methods and the stereo problem,” INRIA, Tech. Rep. 3021, 1996, pp. 1–37.
- [51] O. Faugeras, J. Gomes, and R. Keriven, “Computational stereo: A variational method,” in *Geometric Level Set Methods in Imaging, Vision, and Graphics*, New York: Springer-Verlag, 2003, pp. 343–360.
- [52] A. Yezzi and S. Soatto, “Stereoscopic segmentation,” in *Proc. of Int. Conf. Computer Vision*, Jul. 2001, pp. 59–66.
- [53] ———, “Stereoscopic segmentation,” *International J. of Computer Vision*, vol. 53, pp. 31–43, Jun. 2003.
- [54] G. Gallego, A. Yezzi, F. Fedele, and A. Benetazzo, “A variational stereo method for the three-dimensional reconstruction of ocean waves,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 49, pp. 4445–4457, 2011.
- [55] F. Fedele, G. Gallego, A. Yezzi, A. Benetazzo, L. Calaveri, M. Sclavo, and M. Bastianini, “Euler characteristics of oceanic sea states,” *Mathematics and Computers in Simulation*, vol. 82, pp. 1102–1111, 2012.
- [56] F. Fedele, A. Benetazzo, G. Gallego, P. Shih, A. Yezzi, F. Barbariol, and F. Ardhuin, “Space-time measurements of oceanic sea states,” *J. Ocean Modeling*, special issue: Ocean Surface Waves, vol. 70, pp. 103–115, Oct. 2013.
- [57] G. Gallego, A. Yezzi, F. Fedele, and A. Benetazzo, “Variational stereo imaging of oceanic waves with statistical constraints,” *IEEE Trans. Image Processing*, vol. 22, pp. 4211–4223, Nov. 2013.
- [58] H. Jin, A. Yezzi, and S. Soatto, “Variational multiframe stereo in the presence of specular reflections,” in *Proc. of 3D Data Processing Visualization and Transmission*, Jun. 2002, pp. 626–630.
- [59] H. Jin, A. Yezzi, R. Tsai, L. Cheng, and S. Soatto, “Estimation of 3d surface shape and smooth radiance from 2D images,” *Journal of Scientific Computing*, vol. 19, pp. 267–292, Dec. 2003.
- [60] H. Jin, S. Soatto, and A. Yezzi, “Multi-view stereo beyond lambert,” in *Proc. of Computer Vision and Pattern Recognition*, Jun. 2003, pp. 171–178.
- [61] S. Soatto, A. Yezzi, and H. Jin, “Tales of shape and radiance in multiview stereo,” in *Proc. of Int. Conf. Computer Vision*, 2003, pp. 974–981.
- [62] A. Yezzi, S. Soatto, A. Tsai, and A. Willsky, “The mumford-shah functional: From segmentation to stereo,” in *Image Analysis: Low and High Level Vision (Vol. 133*

in *IMA Volume in Mathematics and its Applications*), New York: Springer-Verlag, 2003, pp. 125–147.

- [63] A. Yezzi, S. Soatto, H. Jin, A. Tsai, and A. Willsky, “Mumford-shah from segmentation to stereo,” in *Geometric Level Set Methods in Imaging, Vision, and Graphics*, New York: Springer-Verlag, 2003, pp. 207–228.
- [64] A. Yezzi and S. Soatto, “Structures from motion for scenes without features,” in *Proc. of Computer Vision and Pattern Recognition*, 2003, pp. 525–532.
- [65] H. Jin, D. Cremers, A. Yezzi, and S. Soatto, “Shedding light on stereo segmentation,” in *Proc. of Computer Vision and Pattern Recognition*, Jul. 2004, pp. 36–42.
- [66] H. Jin, A. Yezzi, and S. Soatto, “Region-based segmentation on evolving surfaces with application to 3d reconstruction of shape and piecewise constant radiance,” in *Proc. of European Conf. Computer Vision*, May 2004, pp. 114–125.
- [67] H. Jin, S. Soatto, and A. Yezzi, “Multiview stereo reconstruction of dense shape and complex appearance,” *Int. Journal of Computer Vision*, vol. 63, pp. 175–189, Jul. 2005.
- [68] H. Jin, A. Yezzi, and S. Soatto, “Mumford-shah on the move: Region-based segmentation on deforming manifolds with application to 3-d reconstruction of shape and appearance from multiview images,” *J. Mathematical Imaging and Vision*, vol. 29, pp. 219–234, Nov. 2007.
- [69] G. Unal, A. Yezzi, S. Soatto, and G. Slabaugh, “A variational approach to problems in calibration of multiple cameras,” *Trans. Pattern Anal. Machine Intell.*, vol. 29, pp. 1322–1338, Aug. 2007.
- [70] H. Jin, D. Cremers, D. Wang, E. Prados, A. Yezzi, and S. Soatto, “3-d reconstruction of shaded objects from multiple images under unknown illumination,” *Int. Journal of Computer Vision*, vol. 76, pp. 245–256, Mar. 2008.
- [71] A. Benetazzo, F. Fedele, G. Gallego, P. Shih, and A. Yezzi, “Offshore stereo measurements of gravity waves,” *Coastal Engineering*, vol. 64, pp. 127–138, Jun. 2012.
- [72] A. Yildirim and A. Yezzi, “Developing a geometric deformable model for radar shape inversion,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 6483–6487.
- [73] M. R. Oswald, “Convex variational methods for single-view and space-time multi-view reconstruction,” PhD thesis, Technical University of Munich, 2014.

- [74] B. K. P. Horn, "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view," PhD thesis, Massachusetts Institute of Technology, Nov. 1970.
- [75] R. Woodham, "Photometric method for determining surface orientation from multiple images," *Optical Engineering*, vol. 19, Jan. 1992.
- [76] A. R. Bruss, "The image irradiance equation: Its solution and application," PhD thesis, Massachusetts Institute of Technology, Jun. 1981.
- [77] B. K. P. Horn and R. W. Sjöberg, "Shape from shading," in B. K. P. Horn and M. J. Brooks, Eds., Cambridge, MA, USA: MIT Press, 1989, ch. Calculating the Reflectance Map, pp. 215–244, ISBN: 0-262-08183-0.
- [78] R. Jain, R. Kasturi, and B. Schunck, "Machine vision," in First. McGraw-Hill, Inc., 1995, ch. 9.
- [79] R. Kimmel and J. A. Sethian, "Optimal algorithm for shape from shading and path planning," *Journal of Mathematical Imaging and Vision*, vol. 14, pp. 237–244, 2001.
- [80] L. Zhang, B. Curless, A. Hertzmann, and S. Seitz, "Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo," in *Proc. of ICCV*, 2003.
- [81] A. Tankus, N. Sochen, and Y. Yeshurun, "Shape-from-shading under perspective projection," *International Journal of Computer Vision*, vol. 63, no. 1, pp. 21–43, 2005.
- [82] C. N. Carter, R. J. Pusateri, D. Chen, A. H. Ahmed, and A. A. Farag, "Shape from shading for hybrid surfaces as applied to tooth reconstruction," in *2010 IEEE International Conference on Image Processing*, Sep. 2010, pp. 4049–4052.
- [83] R. Huang and W. A. P. Smith, "Shape-from-shading under complex natural illumination," in *2011 18th IEEE International Conference on Image Processing*, Sep. 2011, pp. 13–16.
- [84] S. Magda, T. Zickler, D. J. Kriegman, and P. B. Belhumeur, "Beyond lambert: Reconstructing surfaces with arbitrary bdrfs," in *Proc. of ICCV*, 2001, pp. 391–398.
- [85] M. Subbarao and G. Surya, "Depth from defocus: A spatial domain approach," *International Journal of Computer Vision*, vol. 13, pp. 271–294, 1994.

- [86] N. Asada, H. Fujiwara, and T. Matsuyama, “Edge and depth from focus,” *Int. J. of Comp. Vision*, vol. 26, pp. 153–163, 1998.
- [87] A. P. Pentland, “A new sense for depth of field,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 523–531, 1987.
- [88] T. Darell and K. Wohn, “Depth from defocus using a pyramid architecture,” *Pattern Recognition Letters*, vol. 11, pp. 787–796, 1990.
- [89] H. Jin and P. Favaro, “A variational approach to shape from defocus,” in *Proc. of the Eur. Conf. on Computer Vision*, 2002, pp. 18–30.
- [90] P. Favaro and S. Soatto, “3d shape estimation and image restoration: Exploiting defocus and motion blur,” in: Springer, 2007.
- [91] S. Gur and L. Wolf, “Single image depth estimation trained via depth from defocus cues,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 7675–7684.
- [92] M. Carvalho, B. Le Saux, P. Trouvé-Peloux, A. Almansa, and F. Champagnat, “Deep depth from defocus: How can defocus blur improve 3D estimation using dense neural networks?” In *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds., Cham: Springer International Publishing, 2019, pp. 307–323, ISBN: 978-3-030-11009-3.
- [93] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, “Surfacenet: An end-to-end 3d neural network for multiview stereopsis,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2326–2334.
- [94] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, “Pix2vox: Context-aware 3d reconstruction from single and multi-view images,” in *2019 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 2690–2698.
- [95] X. Song, X. Zhao, and L. F. et al., “Edgestereo: An effective multi-task learning network for stereo matching and edge detection,” *Int J Comput Vis*, vol. 128, pp. 910–930, Jan. 2020.
- [96] B. Yang, S. Wang, A. Markham, and N. Trigoni, “Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction,” *Int J Comput Vis*, vol. 128, pp. 53–73, Aug. 2020.
- [97] N. Xu and N. Ahuja, “On the use of depth-from-focus in 3D object modeling from multiple views,” in *Proc. of Asian Conference of Computer Vision*, Jan. 2003, pp. 1038–1043.

- [98] T. Wang, M. Srikanth, and R. Ramamoorthi, “Depth from semi-calibrated stereo and defocus,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3717–3726.
- [99] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, “Depth from combining defocus and correspondence using light-field cameras,” in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 673–680.
- [100] R. D. Friedlander and A. J. Yezzi, “A closed-form expression for thin lens image irradiance,” in *Proc. of 2017 International Conference on Image Processing Theory, Tools, and Applications*, © 2017 IEEE, Nov. 2017.
- [101] R. D. Friedlander and A. J. Yezzi, “Closed-form solution for thin lens image irradiance under arbitrary solid angle,” *J. Opt. Soc. Am. A*, vol. 37, no. 4, pp. 568–578, Apr. 2020, © 2020 The Optical Society.
- [102] B. K. P. Horn, “Robot vision,” in, First. McGraw-Hill Book Company, 1986, ch. 10-11.
- [103] H. Flanders, “Differentiation under the integral sign,” *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.
- [104] I. S. Gradshteyn and I. M. Ryzhik, “Table of integrals, series, and products,” in, Eighth. Elsevier, Oct. 2014, ch. 10.
- [105] A. Gray, E. Abbena, and S. Salamon, “Modern differential geometry of curves and surfaces with mathematica,” in, Second. CRC Press, 1997, pp. 369–371.
- [106] L. Hörmander, “The analysis of linear partial differential equations i: Distribution theory and fourier analysis,” in, Second. Springer, 1990, p. 256.