May 2018

Morphological and Energy Transport Optimization of Spectrally-Selective Solar Absorber Coatings at Mesoscale

Dale Karas
dalekaras@gmail.com

Follow this and additional works at: https://digitalscholarship.unlv.edu/thesesdissertations

Part of the Engineering Science and Materials Commons, and the Materials Science and Engineering Commons

Repository Citation

https://digitalscholarship.unlv.edu/thesesdissertations/3273

This Dissertation is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
MORPHOLOGICAL & ENERGY TRANSPORT OPTIMIZATION
OF SPECTRALLY-SELECTIVE SOLAR ABSORBER
COATINGS AT MESOSCALE

By

Dale E. Karas

Bachelor of Music (BM) - Music Composition
Bachelor of Science (BS) - Optical Sciences & Engineering
The University of Arizona
2013

A dissertation submitted in partial fulfillment
of the requirements for the

Doctor of Philosophy - Mechanical Engineering

Department of Mechanical Engineering
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
May 2018
This dissertation prepared by

Dale E. Karas

entitled

Morphological and Energy Transport Optimization of Spectrally-Selective Solar Absorber Coatings at Mesoscale

is approved in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Engineering - Mechanical Engineering
Department of Mechanical Engineering

Jaeyun Moon, Ph.D.
Examination Committee Chair

Robert F. Boehm, Ph.D.
Examination Committee Member

Evangelos A. Yfantis, Ph.D.
Examination Committee Member

Yi-Tung Chen, Ph.D.
Examination Committee Member

Kathleen Robins, Ph.D.
Graduate College Faculty Representative

Kathryn Hausbeck Korgan, Ph.D.
Graduate College Interim Dean
ABSTRACT

MORPHOLOGICAL & ENERGY TRANSPORT OPTIMIZATION OF SPECTRALLY-SELECTIVE SOLAR ABSORBER COATINGS AT MESOSCALE

by

Dale E. Karas

Jaeyun Moon, Ph.D., Examination Committee Chair
Assistant Professor of Mechanical Engineering
University of Nevada, Las Vegas

A special class of cuprous-based inorganic oxide materials, synthesized as nanoparticles via hydrothermal and co-precipitation methods, are portable to spectrally-selective absorber coatings with high solar-thermal energy conversion efficiency. Operating reliably at elevated temperatures when used in tandem with solar concentrators, these materials enable cost-competitive solar energy conversion technology that can be incorporated with thermal energy storage systems, supporting the viability of novel renewable power generation; notably, optimizing absorptive performance while mitigating thermal losses through re-radiated waste heat motivates sustainable energy production particular to desert climates, where water conservation and ecological sensitivity needs are paramount.

This work targets the chemical synthesis optimization of such absorber coat-
ing materials to reliably form spectrally-selective surface texturing. Specifically, the synthesis of phase-stable uni-metallic and bi-metallic oxide materials (CuO, Co₃O₄, Cu₀.₁₅Co₂.₈₄O₄, Cu₁.₅Mn₁.₅O₄), while viable for bulk manufacturability of absorber coatings, can be improved to increase solar absorptive capability with the addition of embedding sacrificial polymer beads. By modifying coating surface morphology, adjustable porous geometries materialize at mesoscale, enabling facile light-trapping structures for high ultraviolet and visible spectral absorptance while limiting infrared emittance. Morphological detail of the fabricated coating materials, as qualified by Field-Emission Scanning Electron Microscopy (FESEM), determine quantitative correlations in calculating spectral absorptance, optical scatter, and irradiance/emittance distributions of incident solar radiation. Image processing on the material’s microscopy data is used in custom raytracing simulations that calculate energy propagation to correlate material properties based on surface structuring. To ensure an accurate representation of the sample morphology, multiresolution analysis is performed to construct approximated surface profiles of the material. Ultimately, these computational approaches are proposed for optimizing chemical reaction conditions of inorganic nanomaterial syntheses, demonstrating simulation approaches to predict coating performance, supporting the characterization of nanomaterial fabrication that result in absorber coatings tenable for long-term usage in solar power technologies.
ACKNOWLEDGMENTS

“To the everlasting glory of those few men blessed and sanctified in the curses and execrations of those many whose praise is eternal damnation.”
- Kaikhosru Shapurji Sorabji

With a tearful prelude and rigorous attempts to defenestrate my thesaurus so as not to look up multiple ways to use the word ‘support’, I have attempted in large efforts to remember all of you who have inspired me, have nurtured me, who have not only helped me to make it this far, but who will also further help me to continue the adventure. I added to this list starting from the previous year (when it seemed as though authoring a dissertation was actually feasible) – all those many names that would come to me little by little. (If I’ve somehow forgotten to place your name here among the many acknowledgments, please pop me an e-mail and I’ll consider mentioning you as an inspiration source if the Nobel acceptance speeches are not limited to 160 characters in the near future.)

To my [positive adjectives redacted preceding award of doctoral degree] committee members, Dr. Jaeyun Moon, Dr. Robert Boehm, Dr. Evangelos Yfantis, Dr. Yi-Tung Chen, and Dr. Kathleen Robins – my utmost gratitude for the years of research guidance, inspiration, and patience.

To my superlative funding sources, financial supporters, and benefactors: Solar-
Water-Environmental NEXUS in Nevada administrators Dr. Gayle Dana, Michele Casella, and Jane Palmer, support in part by the National Science Foundation (NSF) Grant #IIA-1301726, Photon Engineering, LLC for the University Gratis License of FRED Optical Engineering Software, and Dr. Jack D. Gaskill and Dr. Sean McCafferty, for whom I may never have persevered in research and academia.

To my wonderful undergraduate mentees who supported me and many aspects of this work, Cilla Jose, Astrid Miller, Idalia Soto, Samuel Tam – my love and best wishes to you all, as well as to my other fellow group mates in the Energy & Environmental Materials Laboratory (EEML), for the camaraderie, unceasing encouragement, and frequent alerts of my committee’s whereabouts so as to successfully complete this dissertation in a timely manner: Dr. Hyeunhwan An, Dr. Jongmin Byun, Soroosh Mortazavian, Matt Pusko, Charles Han, and David Zagaceta.

To my collaborators at the University of California San Diego, Dr. Renkun Chen and Dr. Elizabeth Caldwell, who assisted with experimental verifications and provided exceptional recommendations for our research progress.

To my institution’s additional faculty/staff, whose valuable mentorship helped with all the numerous little things: Dr. Minghua Ren, Dr. Ashkan Salamat, Dr. Thomas Hartmann, Dr. Erica Marti, Dr. Ke-Xun Sun, Ronald Young, Julie Longo, Sue Wainscott, Katherine Keller, Dr. Brenden O’Toole, Dr. Mohamed Trabia, Joan Conway, Valarie Burke, Katelyn DiBenedetto, and Dr. Kate Korgan.

To the many non-committee member doctors, former advisors, and mentors who held nothing back: Dr. Dimitri Kececioglu, Dr. Stephen M. Kuebler, Dr. Ananthakr-
ishnan Narayanan, Dr. J. Scott Tyo, Dr. Israel J. Vaughn, Dr. Russell Chipman, Dr. R. John Koshel, Dr. Wolfgang Fink, Dr. John Greivenkamp, Dr. Michael Nofziger, Dr. Paul Blowers, Dr. Iwona A. Palusinski, Dr. Blake C. Jacquot, Dr. Hubert “Buddy” Martin, Dr. Ming C. Wu, Dr. Niels Quack, Dr. Michael Dauphinais, Dr. Mykola Suk, Dr. Aden Meinel, and Dr. S. J. Ben Yoo.

To my other valuable friends and colleagues in optics, Rich and Donata Pfisterer, Ryan Irvin, Robert Buckles, Trin Riojas, Kristin Waller, Geneviève Gish Allouche, R. Dawson Baker, and Emily Finan.

To those supporting professional societies and research organizations comprising my research affiliations: The International Society of Optics & Photonics (SPIE), The Optical Society (OSA), the Materials Research Society (MRS), the American Chemical Society (ACS), the Institute of Electrical and Electronics Engineers (IEEE), the American Nuclear Society (ANS), the American Society of Mechanical Engineers (ASME), the Autodesk Developer Network (ADN), and the American Hellenic Educational Progressive Association (AHEPA).

To my lifelong friends, colleagues, and collaborators, Norris Lam, Alexander Lacson, Dr. Suzanne Rich, Alan Campbell, and Victor Matthews, for their unremitting honesty, kindness, laughter, generosity, and loyalty.

To my wonderful parents, for their love and support, who would only occasionally bother me about my research progress and feign interest in this work.

...et à Calico Alexandra, qui veillait sur moi.

vii
DEDICATION

To Dale E. Karas. I write for me.
PREFACE

“A thesis has to be presentable... but don't attach too much importance to it. If you do succeed in the sciences, you will do later on better things and then it will be of little moment. If you don't succeed in the sciences, it doesn't matter at all.”
- Paul Ehrenfest

This dissertation details the experimental development and refinement of specialized intrinsic absorber coatings for solar-thermal energy conversion, contributing to the development of solar energy technologies, especially for concentrated solar power (CSP) systems. With the project motivation of boosting the efficiency of power generation technologies and limiting the required water usage necessary for operations, especially in desert climates, the principle project work herein spanned a multi-level research consortium at The University of Nevada, Las Vegas (UNLV), The University of Nevada, Reno (UNR), and Desert Research Institute (DRI), supporting research in solar power generation, water treatment, environmental impact, economics of energy factors, and novel cyber-infrastructure governing security and control of such sustainable living technologies. Primarily, the experimental development and characterization of solar absorber coating materials was performed at UNLV’s Energy & Environmental Materials Laboratory (EEML) under Dir. Jaeyun Moon, Ph.D. Simulation aspects of analysis and optimization of experimental data, in providing a synthesis workflow, was performed at the UNLV National Supercomputing Insti-
tute (formally NSCEE, the National Supercomputing Center For Energy and the Environment).

The Nevada Solar-Water-Environmental NEXUS  Nevada has been recognized to great economic detail of providing solar technologies to help drive the state’s power needs, placed in a geographical climate with high solar irradiation. Yet, nearby land locales such as the Great Basin and Mojave Desert both contain fragile ecosystems easily altered by anthropogenic activities. Additional considerations concern the water usage required to operate most types of power plants in such arid regions of the United States. As described from the Solar Energy-Water-Environment NEXUS core mission:

“Started June 1, 2013, the Solar Nexus Project (for short) is a multifaceted five-year research project focusing on the nexus (or linkage between) solar energy generation and Nevada’s limited water resources and fragile environment. The focus of the Solar Nexus Project is creating a center of research excellence on solar energy conversion to electricity, minimizing its negative impacts on water usage and the environment. In essence, seeking to create a paradigm shift in how solar plants are built and utilized, helping Nevada establish itself as a competitive state in the field of solar nexus research.”

This dissertation largely researches the efforts of improving novel and sustainable solar power generation technologies, which support desert ecosystem protections and water conservation needs.

Solar Absorber Coating Material Development

An array of materials produced in the Solar-Water-Environmental NEXUS in Nevada program range from energy harvesting devices, novel materials used in water
treatment and purification technologies, and improvements to next-generation solar production in such southwestern United States desert climates.

Particularly to materials in the sector of solar power, two main technologies exist – photovoltaic systems (PV) and Concentrates Solar Power Systems (CSP). The latter CSP, a renewable energy source driving conventional steam turbines and engines through concentrating the sun’s energy with mirror systems (heliostats), has been of considerable interest for energy-efficient research, as Department of Energy (DoE) projections speculate that it will become an important component of the energy portfolio in the United States within the next 50 years. To reach sustainable energy production targets that are cost-competitive with the projected costs of compound photovoltaics, current materials research for CSP systems is largely concerned with the development of novel specialized absorber coatings that improve upon solar-thermal conversion efficiency metrics. Short-term needs for reliable, maintainable, and cost-effective coatings have largely motivated the following studies herein for the development of solar absorber materials with enhanced solar-thermal conversion efficiency.

**Research Question #1**: Will intrinsic absorber coatings become amenable to CSP solar-thermal energy conversion technologies so as to improve cost-competitiveness in their market structure? Ensuring cost-effectiveness of CSP platforms is enabled through higher temperature operation, maximizing electric power output from solar thermal energy conversion. One significant technology
for reliable high-temperature operation has been the implementation of high temperature durable spectrally-selective solar absorber coatings – they can absorb ultraviolet, visible, and near-infrared solar irradiation while limiting spontaneous thermal radiation from emittance at higher wavelengths. Fabricating such coatings relies on an optimization for low-cost synthesis while allowing high-temperature operation and spectrally-selective behavior.

Simulation of Material Transport at Mesoscale

Succeeding the synthesis, fabrication, and deposition of material coatings on a corrosive-resistant substrate capable of withstanding high temperatures, adapting a computational model to combine experimental analyses and help qualify refinements to the chemical synthesis was prudent. Despite a variety of established methods for simulating material performance factors at larger scales, few robust methods exist to model photon transport in mesoscopic structures. Primarily, with the application of computational methods used in optical design, the following study in correlating surface topology with light-matter interaction was posed.

Research Question #2: Based on surface modifications to the sample morphology, what can be inferred in correlating energy transport to structuring? Principles of material science consistently relate material structuring to their properties and performance. Theoretical models, as well as the ease and cost of experimental testing, for material performance at larger macroscopic scales, generally
provide a sufficient description of interdependent mechanical, thermal, and acoustic properties. Recent instrumentation permitting the fabrication and analysis of nanotechnology require atomic-scale simulation methods for determining formation and predicting operating behaviors, for which their developments leverage cataloguing environmental impacts.

**Simulation Approaches** Monte Carlo methods, while computationally expensive, provide an avenue to classify probability distributions of energy transport. In classifying solar absorber coatings, custom raytracing routines are implemented according to a Monte Carlo formulation. Proof-of-concept for raytracing algorithms are written in MATLAB, with a successive C++ port providing recommendations for accelerating computational runtime.

The extent of results on a variety of experimentally-derived solar absorber samples are assessed with microscopy data. Image processing routines, governed by a linear systems formulation, are developed to estimate additional factors of fabricated coatings – energy transport based on successive specular reflection approximate large probabilities of scatter events.

**Statistics of Morphological Optimization**

To perform a thorough analysis on the developed simulation runtime, the following final study is conducted:
Research Question #3: Are there statistical optimization methods for determining improvements to coating surface morphology, for spectrally-selective absorptance quality? While the raytracing methods developed in the second study employ direct specularly-reflective interactions when interfacing with the material, a linear systems formulation based on interpreting porosity from image processing techniques perfects the simulation capability of determining spectral absorptance quality. Using techniques from multiresolution analysis, tolerancing suspected errors of microscopy data can be interpreted and compared to theoretical models that characterize optimal light-trapping structures, based on a class of special functions that approximate porous morphology amenable to specific chemical reaction conditions.

Implications of the conducted studies are intended to characterize short-term needs for CSP absorber coating technologies, supporting the synthesis and fabrication of intrinsic, spectrally-selective coatings that can retain optimal mesostructures at elevated temperatures. Additionally, improvements to the runtime of the developed simulations can support rapid prototyping and production of novel inorganic coating materials for energy-efficient applications. A thorough characterization and catalogue of materials formalized through mesoscale simulation can ultimately compare to securing viable energy technologies, nullifying products that have toxicological concern, and supporting reliable production technologies concomitant with ensuring sustainable ecosystems based on applicable technologies.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>viii</td>
</tr>
<tr>
<td>PREFACE</td>
<td>ix</td>
</tr>
<tr>
<td>Solar Absorber Coating Material Development</td>
<td>x</td>
</tr>
<tr>
<td>Simulation of Material Transport at Mesoscale</td>
<td>xii</td>
</tr>
<tr>
<td>Statistics of Morphological Optimization</td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xvii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xx</td>
</tr>
<tr>
<td>LIST OF ALGORITHMS</td>
<td>xxi</td>
</tr>
<tr>
<td>LIST OF LISTINGS</td>
<td>xxiii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Solar Energy &amp; Concentrated Solar Power</td>
<td>1</td>
</tr>
<tr>
<td>Spectrally-Selective Coatings</td>
<td>4</td>
</tr>
<tr>
<td>Fabrication of Novel Intrinsic Absorbers</td>
<td>6</td>
</tr>
<tr>
<td>Simulation Approaches for Experimental Optimization</td>
<td>7</td>
</tr>
<tr>
<td>Summary of Contents</td>
<td>9</td>
</tr>
<tr>
<td>CHAPTER 2 ENERGY PROPAGATION &amp; TRANSPORT</td>
<td>12</td>
</tr>
<tr>
<td>Radiometry &amp; Optical Propagation</td>
<td>14</td>
</tr>
<tr>
<td>Heat Transfer</td>
<td>24</td>
</tr>
<tr>
<td>Solar Energy Transport</td>
<td>25</td>
</tr>
<tr>
<td>Diffraction &amp; Optical Scattering</td>
<td>31</td>
</tr>
<tr>
<td>Statistical Summary</td>
<td>34</td>
</tr>
<tr>
<td>Multiresolution Analysis</td>
<td>36</td>
</tr>
<tr>
<td>CHAPTER 3 EXPERIMENTAL SYNTHESSES</td>
<td>43</td>
</tr>
</tbody>
</table>
LIST OF TABLES

2.1 Table of infrared wavelength ranges, near infrared (NIR), short-wave infrared (SWIR), mid-wave infrared (MWIR), long-wave infrared (LWIR), and far infrared (FIR). ................................................................. 14
2.2 Table of radiometric quantities ......................................................... 14

3.1 Mid-range temperature cermet material properties, reproduced from Cao et. al. [1] ........................................................................................................ 45
3.2 High-temperature inorganic material properties in oxide matrix, reproduced from Bermel et. al. [2] and Chester et. al. [3] ................................. 45
3.3 Listed high-temperature solar selective coatings: single-, double-layered cermets; multilayer and tandem absorbers [4] ................................. 46
3.4 Chemical reaction detail of precursor inorganic chlorohydrates resulting in desired cuprous spinel nanomaterials ........................................ 49
3.5 Cu_{0.15}Co_{2.84}O_4 and Cu_{1.5}Mn_{1.5}O_4 reaction detail, porting nanopowders to a viscous slurry solution capable for spray-coating. .......... 53
3.6 Figure-of-Merit (FoM) efficiency calculated for fabricated absorber coatings ........................................................................................................... 78

4.1 Table outlining the relationships of inputs and outputs properties for different material scaling. (Reproduced in part from Pollock et. al. [5]) 83
## LIST OF FIGURES

1.1 Concentrated Solar Power (CSP) hardware: heliostats and solar receiver components .......................................................... 3

1.2 Total CSP System Schematic, reproduced from Mehos et. al. [6] .... 5

2.1 Spectral considerations for absorptive materials dependent on CSP operating temperature and concentration factor. Graph depicts ASTM-G173 solar irradiation profile [7] and normalized Planck radiance of a blackbody at \( T = 700 \) °C ................................................................. 27

2.2 Operating efficiency of concentrated solar power systems with recommended concentration factor based on the absorber coating’s reliable operating temperature .......................................................... 30

2.3 Description of tungsten-halogen source used for experimental spectral absorptive measurements ......................................................... 33

2.4 Visualization of fourier transform pairs, comparing continuous-time (columns 1,2) and discrete-time (columns 3,4) signals .................. 37

2.5 Visualization of wavelet transform methods, leveraging position-frequency or time-frequency content according to varying scale parameters, reproduced from Mallat et. al. [8] ................................. 40

3.1 Precursor inorganic chlorohydrates, such as 1 M CuCl₂·2H₂O, prepared in 1M concentrations for co-precipitation preparation .... 50

3.2 Inorganic oxide nanoparticles placed in a crucible, preceding \( t = 5 \) h annealing treatment for inducing phase stability ...................... 51

3.3 Spray-coated \( \text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4 \)- and \( \text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4 \)-type samples onto Inconel-625 substrates undergoing curing ................................. 54

3.4 \( \text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4 \)-type coating samples on different sizes of Inconel-625 coupons prepared for experimental analysis .............................. 55

3.5 Field-Emission Scanning Electron Microscopy (FESEM) used in this study (JEOL JSM-6700F) ..................................................... 57

3.6 Sample preparation of \( \text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4 \) nanoparticles, prepared on a platform for SEM imaging (left) and sputter-coated with gold for ensuring image capture (right) ................................. 58

3.7 CAD detail for optical absorptance setup, consisting of a tungsten halogen source, scan monochromator, integrating sphere, and photodiode detectors ............................................................. 63
3.8 Field-emission scanning electron microscopy (FESEM) images comparing both Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ cuprous-based nanoparticles.

3.9 FESEM of Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ solar absorber coating materials.

3.10 XRD of cuprous-based nanoparticles.

3.11 XRD of cuprous-based solar absorber coatings.

3.12 Optical Data.

3.13 Energy Dispersive X-Ray Spectroscopy (EDS) of Coatings Materials (Pyromark 2500 and developed Cu$_{1.5}$Mn$_{1.5}$O$_4$ coating).

3.14 Coating Figure-of-Merit.

4.1 Flow diagram of experimental and simulation processes to optimize spectrally-selective solar absorber coatings based on energy transport phenomenology from continuously-refined surface texturing.

4.2 Image data, optical and field-emission scanning electron microscopy.

4.3 Tessellated SEM image for Monte Carlo data calculations, expanding the usable data runtime according to rays that diverge immediately past the boundary confined by the central 968x1280 dataset.

4.4 Monte Carlo Raytracing Visualization.

4.5 Monte Carlo Raytracing Summary: Ray Booleans.

4.6 Monte Carlo Raytracing Summary: Ray Powers.

4.7 Monte Carlo Raytracing Summary: Ray x-Directions.

4.8 Monte Carlo Raytracing Summary: Ray y-Directions.

4.9 Monte Carlo Raytracing Summary: Ray z-Directions.

4.10 Monte Carlo Raytracing Summary: Ray x-Offset.

4.11 Monte Carlo Raytracing Summary: Ray y-Offset.

4.12 Monte Carlo Raytracing Summary: Ray z-Offset.

4.13 Monte Carlo Raytracing Summary: Ray Azimuths.

4.14 Monte Carlo Raytracing Summary: Ray Elevations.

4.15 Monte Carlo Raytracing Summary: Additional Raytrace Data.

4.16 Raytracing Summary on Prominent FESEM Images.

5.1 2D cross section of material: signal smoothing by noise removal using wavelet transform methods.

5.2 Histogram equalization routine, demonstrating the changes incurred on the SEM image for improving feature extraction.

5.3 Equivalent surface plots, with RGB values in z.

5.4 Equivalent surface plots, scaled in z for surface height.

5.5 Surface comparisons to 3D Laser-Scanning Confocal Microscopy Data.

5.6 Porosity histograms of standard Cu$_{0.15}$Co$_{2.84}$O$_4$- and Cu$_{1.5}$Mn$_{1.5}$O$_4$-based solar absorber coatings, as calculated through spatial-domain scanning of x- and y-cross sections, and averaged to a Gamma distribution.
5.7 Porosity histograms of SPB-embedded Cu$_{0.15}$Co$_{2.84}$O$_4$- and Cu$_{1.5}$Mn$_{1.5}$O$_4$-based solar absorber coatings, as calculated through spatial-domain scanning of x- and y-cross sections, and averaged to a Gamma distribution. ........................................... 119

5.8 For nanoparticles present on a substrate, the Atherton-Kerbyson modification of the circular Hough Transform can track particle locations and sizes. .......................................................... 121

5.9 Standard and surface-textured Cu$_{0.15}$Co$_{2.84}$O$_4$ comparing histograms before and after applying image thresholding ................. 122

5.10 Watershedding Algorithm on a Cu$_{0.15}$Co$_{2.84}$O$_4$ sample ............... 124

5.11 Array of Spherical, Trigonal, and Power Functions used for a data reference to compare frequency content of microscopy data ........ 125

5.12 Array of Spherical, Trigonal, and Power Functions used for a data reference to compare frequency content of microscopy data ........ 126

5.13 Pore Size Histogram ..................................................... 127

5.14 Experimentally-derived spectral absorptance response of Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ ................................................. 128

5.15 Spectral absorptance response of Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$, as derived from simulation methods .................. 129
LIST OF ALGORITHMS

1  Brute-force raytracing runtime, compacting listings as presented in App.1  88
LIST OF LISTINGS

6.1 Main raytracing runtime, including function calls ....................... 136
6.2 Calculate surface normals at all sampled points ...................... 138
6.3 Nested function call to calculate singular surface normal based on
neighboring pixel intensities ........................................... 139
6.4 Calculate initial reflected rayset array based on incident propagation
vector ................................................................. 140
6.5 Determine successive ray-to-surface interaction ....................... 140
6.6 Determine new normal vector at an interpolated surface point ....... 140
6.7 Calculate singular reflected ray vector based on incident propagation
ray and interpolated surface normal vector ................................ 141
6.8 Main raytracing runtime via C++ port: Header File .................. 141
6.9 Main raytracing runtime via C++ port: C++ File ..................... 141
6.10 Ray Normals Calculation: Header File ................................ 142
6.11 Ray Normals Calculation: C++ File .................................. 142
6.12 Calculate Cross-Section Normals: Header File ....................... 146
6.13 Calculate Cross-Section Normals: C++ File ......................... 147
6.14 Calculate Specular Reflection Array: Header File .................... 149
6.15 Calculate Specular Reflection Array: C++ File ...................... 149
6.16 Calculate Specular Reflection Vector: Header File .................... 150
6.17 Calculate Specular Reflection Vector: C++ File ...................... 151
6.18 Calculate Specular Reflection Vector: Header File .................... 151
6.19 Calculate Specular Reflection Vector: C++ File ...................... 152
6.20 Calculate Interpolated Normal: Header File .......................... 153
6.21 Calculate Interpolated Normal: C++ File ............................. 153
CHAPTER 1
INTRODUCTION

“Available energy is the main object at stake in the struggle for existence and the evolution of the world.”
- Ludwig Boltzmann

This work details research motivation in improving energy conversion efficiency for a class of inorganic oxide materials [6], specifically for solar power generation that is a competitive technology for the southwestern United States [9]. Particularly to materials in the sector of solar power, two main technologies exist - photovoltaic systems (PV) and Concentrated Solar Power (CSP) systems [10]. CSP, a renewable energy source driving conventional steam turbines and engines through concentrating the sun’s energy with mirror systems, has been of considerable interest for energy-efficient research, as Department of Energy (DoE) projections [11] speculate that it will become an important component of the energy portfolio in the United States within the next 50 years.

Solar Energy & Concentrated Solar Power

Concentrated Solar Power (CSP) technologies compare to photovoltaics by introducing the intermediary step of converting energy to heat, generating electricity by
conventional operation of a heat engine, such as a Rankine or Brayton thermodynamic cycle [12, 13]. With a large array of movable mirrors (heliostats) that track solar positioning, heat is concentrated onto a central receiver – metallic tubular structures that contain a heat transfer fluid such as molten salts [14]. Energy is capable of being stored and transferred to a steam turbine to generate electricity [15].

While CSP technologies cannot be consumer-implemented as with the smaller size requirements for compound photovoltaics, construction of CSP facilities features the following advantages [16, 17]:

- Provides continuous energy operation at times of limited solar irradiance, as energy is able to be stored rather than operating directly based on the photoelectric effect [10]

- Permits adjustable power generation settings given the requirement for large-area installations, supplying variable commercial power needs [18]

- Capable of retrofitting to older, non-renewable power installations such as coal or natural gas plants, as the technology employs conventional energy generation cycles [19]

In Fig.1.1, major components of the CSP energy conversion process are depicted from the Ivanpah Solar Power Facility in Primm, Nevada, depicting heliostats (left) and central receiver tower (right); the receiver is where solar absorber coatings are deposited, supporting thermal energy conversion [20]. With roughly 175,000 heliostats, the installation produces $P = 950 \text{ GW h}$ annual power – at the time of its construc-
tion in 2014, it was the largest solar thermal power plant in operation (soon to be superseded by the 2018 Aurora Plant constructed in South Australia) [21].

Figure 1.1: Concentrated Solar Power (CSP) hardware: heliostats and solar receiver components

To improve the quality of energy production of CSP plants and ensure their cost-competitive manufacturability, operability, and maintainability, a major research impetus suggests improving efficiency metrics of solar-thermal energy conversion [22]. Specifically, on the solar receiver systems (such as the ‘power tower’ configurations as used at the Ivanpah Solar Power Facility), a class of special absorber coatings are applied to the metallic solar receivers; Study.1 of this work investigates chemical
syntheses for bulk-scale absorber coating material manufacturability and streamlined fabrication techniques for facile deposition methods, improving upon state-of-the-art commercial coating performance. Specific economic targets for consideration take the levelized cost of energy (LCoE) metric into account, a contemporary reference account for system-wide solar construction, manufacturing, and operating costs [23].

Spectrally-Selective Coatings

Coating configurations range from materially-homogeneous intrinsic coatings or multilayer structures [24], ceramic-metal composites (cermets) [3], and semiconductor materials [4]. Intrinsic absorbers are the easiest type to manufacture [25, 26], and were predominantly investigated in this study as a cost-effective approach to meet the short-term needs of competitive CSP energy production [27].

To provide for maximum energy-efficiency, solar collectors can take advantage of spectrally-selective coatings [28], or interfaces upon the connection of two material media that allow/prohibit the transfer of specific radiative spectral regions. As depicted in Fig.1.2, the central receiver used in the CSP configuration contains the deposited absorber coating, which must be survivable to elevated working temperatures, such as $T \geq 500^\circ C$. 

Intrinsic coating types, do not easily feature spectrally-selective behavior – i.e. the capability for absorptance of most solar irradiance while rejecting longer wavelengths that reradiate waste heat. Yet, if the materials can survive high solar concentration factors, needs for strict spectral-selectivity are effectively nullified due to minimizations of thermal energy loss.

To recapitulate, for effective solar-thermal energy conversion efficiency, absorber coatings targeted for synthesis in the following studies must successfully feature:

1. High-temperature operation for greater energy-conversion efficiency [6, 29]

2. Cost-competitive standards of manufacturing (hydrothermal synthesis and co-
precipitation methods [30] with facile approaches to surface texturing [31] are implemented in this study for upscaling to larger manufacturability)

3. Tendency to spectrally-selective behavior based on CSP operating conditions [32, 33], where power generation scenarios may require elevated operating temperatures and low concentration factors [1, 2, 34] (fewer necessary heliostats in the CSP system for cost-efficiency)

Fabrication of Novel Intrinsic Absorbers

Current commercial coatings available include solar absorber paints with high absorptance properties. In particular, Pyromark® 2500, a standard absorber coating and reference material generally in operation by CSP plants, requires re-application semi-annually, operates at $T = 500\, ^{\circ}C$, and is not a spectrally-selective material [35].

For the initial experimental studies conducted, the utility of intrinsic absorber coatings includes surface-texturing enhancements with changes to sample surface morphology based on reaction conditions [36]. Light-trapping structures that survive higher CSP operating temperatures are of value to improve tendencies to spectral-selectivity. Such fabricated coating demonstrates high UV-VIS solar absorptance and minimal IR thermal emittance achievable with high CSP operating temperatures and solar concentration factors.
Simulation Approaches for Experimental Optimization

Succeeding the experimental development of novel solar absorber coating materials that are favorable for near-term competitiveness of CSP platforms, custom simulation routines are used to seek improvements to the synthesis so as to characterize the material with further energy-efficient properties. As many reaction mechanisms are deterministic in nature, Monte Carlo methods were employed to track energy propagation extent through the material. Statistical refinements and interpretations are made to evaluate the application of the method to inorganic oxide types of materials.

**Monte Carlo Raytracing Methods**  Monte Carlo simulations are applied to the morphology of inorganic oxide surface detail, tenable with high-resolution images of rough surfaces from optical and field-emission scanning electron microscopy (FE-SEM) – standard metrology that enable critical dimensioning of surface profiles and topological detail [37, 38].

Li et. al. [39] reviews the development of a Monte Carlo simulation method for assessing features of scanning electron microscopy (SEM) images of rough surfaces, with interpretations to electron scattering. Ultimately, quantitative estimations based on known error tolerances provide a description of surface roughness and scattering conditions of energy transport. Ganesan et. al. [40] confirms the validity of geometrical optical energy transport in providing a description of visible and near-infrared optical properties of ceria ceramics, the class of intrinsic solar absorber coating materials developed in early reaction stages. By introducing effective transport properties of the
material media, anisotropy conditions are formalized based on continuum mechanics. Buhr et. al. [41] characterized nanoparticles by scanning electron microscopy in transmission mode, interpreting pore diameters and a set of image processing routines to extract particle sizing and shape. Monte Carlo methods used in this study simulate elastic and inelastic electron scattering from detected particle interactions. Error tolerances in the instrument are levied based on bright- and dark-field modes with absolute image contrasts, comparing to profilometry. This study also shows the utility for Monte Carlo methods to be used in surface-penetrating transmissive energy transport.

**Light-Trapping Structures**  For statistically determining mesoscale energy transport in inorganic oxide materials based on developed raytracing methods, image processing routines for determining porosity are employed for comparison against the performance of theoretical light-trapping structures, well-reviewed due to their ubiquity for compound photovoltaics [42, 43, 44]. Antireflection coatings (titanium dioxides, silicon nitrides, etc.) are not seen as cost-effective for bulk coating manufacturability, as there are still sufficient absorptance losses that need not detract from intrinsic coating properties [45].

Iyengar et. al. [46] investigates light-trapping properties in silicon – chemical etching techniques produce randomly distributed surface structures, comparable to the sacrificial polymer beads used in the reaction of of the cuprous spinel-type coatings developed in the premier study. Gjessing et. al. [47] compares types of periodic
light-trapping structures in thin crystalline silicon solar cells. By comparing absorptance properties of bi-periodic structures with a square lattice (in simulation, this is regarded as a rectangular convolution of a two-dimensional Dirac comb), Gjessing motivated a study for submitting target images of theoretical porous geometries, as used in Study.3 of this work. Collectively, Yu et. al. [48] defines fundamental limits associated with light-trapping grating structures. These topics are discussed rigorously in the theoretical foundations of energy transport calculations used in this work, where the extent of relating computational runtime from geometrical optics must feature wave-dependent interactions for ensuring accurate modeling of optical phenomena [49, 50, 51].

Summary of Contents

The scope of this dissertation will explore energy transport calculations for mesoscale modeling of nanomaterials used in solar power applications. Typically, this can include inorganic oxide-based solar materials such as cuprospinsels (CuFe$_2$O$_4$, concentrated solar power applications) and perovskites (CaTiO$_3$, photovoltaics applications), given their stability at ambient pressures and temperatures and photon harvesting properties. In associated projects spanning this research, specific cuprous-oxide nanomaterials were synthesized based on their solar energy absorptance potential – such coatings are spectrally-dependent radiative absorbers that are especially useful for solar energy technologies, as they are able to absorb ultraviolet, visible, and near-infrared solar irradiation while limiting spontaneous thermal radiation from
emittance at higher wavelengths (suppressing waste heat).

For increasing overall system efficiencies that contain absorber coatings, solar absorbers can operate at higher temperatures and solar concentration factors, maximizing electric power output from solar thermal energy conversion – especially useful in the context of concentrated solar power systems. For this reason, computational approaches that calculate synthesis optimizations are of value in achieving maximum effective solar-thermal energy conversion, while ensuring reliability, maintainability, and sustained high temperature operability of the coatings. As these nanomaterials eventually stabilize to mesoscale scaling when ported to coatings, considerations differentiate between atomic-scale calculations through computational chemistry ab-initio methods and macroscopic modeling approaches that are usually performed through finite-element methods. Generally, modeling transport phenomena at mesoscale usually requires phase-field or Monte Carlo methods to determine material structure. In accounting for light-matter interactions, nonlinear dynamics that take increasing precedent based on material degradation will generally require specialized optical modeling to determine photon propagation based on frequency-mixing and spatial-dependent path differences.

Chapter 2 reviews mathematical preliminaries for radiometric considerations of electromagnetic field propagation, spectrally-dependent heat flux, linear systems theory, and image processing routines apposite for aspects of mesostructural energy transport phenomena.

Chapter 3 details the chemical synthesis and material fabrication of two novel
solar absorber coatings, ported from a class of cuprous spinel-type nanoparticles from precursor chlorohydrates.

Chapter 4 proposes a custom simulation method for calculating specular transport for solar energy, irradiant on the absorber coating surface. Full spectral ray paths are described in detail, based on their interaction with varying degrees of surface roughness inherent in the fabricated coating material. Morphological formations and general topology are supplied according to varying magnifications of microscopy data, and image processing routines are implemented to derive a general description of material porosity, correlating spectral dependence based on a linear systems formulation.

Chapter 5 investigates a series of basis functions as descriptors to providing further analysis to simulation routines derived to assess the spectral absorptance quality of the solar coating materials. Implementing multiresolution analysis provides for spatial optimization of porous geometries, with the motivation of recommending topological detail that can further improve the coating performance, if it is survivable at elevated CSP operating temperatures.

Ultimately, this work hypothesizes, as material structuring can infer properties, analysis by simulation of coating morphological data can infer the extent of energy propagation based on approximated optical scatter, supporting accelerated intrinsic absorber coating synthesis, fabrication, and characterization overall.
Electromagnetic (EM) fields, accounting for the ubiquity of many forms of energy generated from sources, especially those capable of inducing heat sources, includes specific regions of interest for solar irradiation: specifically ultraviolet, visible, and infrared radiation (also denoted as shortwave, longwave, and ambient radiation) [52]. Light is produced by accelerated electromagnetic charges, creating electric and magnetic fields at constant velocity that do not have the ability to propagate to extended distances in space [53].

Absorptance \( \alpha \), reflectance \( \rho \), and transmittance \( \tau \), in particular, are ratios that describe incident radiation on a specific body, either absorbed, reflected, or transmitted respectively. Collectively, they sum to unity in Eq.2.1.

\[
\alpha(\lambda) + \rho(\lambda) + \tau(\lambda) = 1
\] (2.1)

Spectral absorptance \( \alpha(\lambda) \) is comparable to emissivity \( \epsilon(\lambda) \), but are distinguished according to the following definitions [52]:

\[
\alpha(\lambda) \quad \text{and} \quad \epsilon(\lambda)
\]
• Absorptance ($\alpha$) is defined as the fraction of incident light of a given wavelength that is absorbed when light strikes an absorbing surface. For selective surfaces, this can be interpreted as the average of the absorptance at each wavelength weighted according to the intensity distribution with wavelength of sunlight.

• Emittance ($\epsilon$) is defined as the fraction of the emittance of a perfect blackbody at a given wavelength emitted by a heated surface. For selective surfaces, it is considered the average of the emittance at each wavelength weighted according to the blackbody intensity distribution with wavelength at the surface temperature. This is characteristically defined to be prominent in the spectral region of $\lambda = [0.3 : 20]\mu$m.

Kirchoff’s Law describes the conditions for emissivity $\epsilon$ equal to absorptance $\alpha$ in the case of null transmittance (Eq.2.2), where:

$$\epsilon(\lambda) = \alpha(\lambda) = 1 - \rho(\lambda)$$  \hspace{1cm} (2.2)

describing material graybodies for $\lambda$-independent cases and material blackbodies for $\alpha = 1$ cases. This is specific to many regions of infrared radiation defined in Table.2.1, where specified region names subtend different wavelength ranges, associated frequencies, and photon energies.
### Table 2.1: Table of infrared wavelength ranges, near infrared (NIR), short-wave infrared (SWIR), mid-wave infrared (MWIR), long-wave infrared (LWIR), and far infrared (FIR).

<table>
<thead>
<tr>
<th>Region</th>
<th>Range (µm)</th>
<th>Frequency (THz)</th>
<th>Photon Energy (meV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIR</td>
<td>0.75–1.4</td>
<td>214-400</td>
<td>886-1653</td>
</tr>
<tr>
<td>SWIR</td>
<td>1.4-3</td>
<td>100-214</td>
<td>413-886</td>
</tr>
<tr>
<td>MWIR</td>
<td>3-8</td>
<td>37-100</td>
<td>155–413</td>
</tr>
<tr>
<td>LWIR</td>
<td>8–15</td>
<td>20-37</td>
<td>83–155</td>
</tr>
<tr>
<td>FIR</td>
<td>15–1,000</td>
<td>0.3-20</td>
<td>1.2–83</td>
</tr>
</tbody>
</table>

Radiometry & Optical Propagation

Radiometry, techniques of EM radiative measurement, include pyrometry (adaptations to measuring heat flux), photometry (ophthalmic energy measurement), and other spectrally-dependent considerations when determining total energy transport, or optical throughput from sources to detection points. Energy metrics, with a nomenclature predominantly used in this work, is given in Table 2.2, distinguishing power (energy per unit time), intensity (energy subtended per solid angle), irradiance/exittance (energy per unit area), and radiance (energy per unit area per unit solid angle).

### Table 2.2: Table of radiometric quantities

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Symbol</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>$Q$</td>
<td>$-$</td>
<td>J</td>
</tr>
<tr>
<td>Power/Flux</td>
<td>$\Phi$</td>
<td>$dQ/dt$</td>
<td>W</td>
</tr>
<tr>
<td>Intensity</td>
<td>$I$</td>
<td>$d\Phi/d\omega$</td>
<td>W/sr</td>
</tr>
<tr>
<td>Exitance</td>
<td>$M$</td>
<td>$d\Phi/dA$</td>
<td>W/m²</td>
</tr>
<tr>
<td>Irradiance</td>
<td>$E$</td>
<td>$d\Phi/dA$</td>
<td>W/m²</td>
</tr>
<tr>
<td>Radiance</td>
<td>$L$</td>
<td>$d^2\Phi/dAd\omega \cos \theta$</td>
<td>W/m²sr</td>
</tr>
</tbody>
</table>

Table 2.2: Table of radiometric quantities
Ray Theory  Approaches as early as 16th-century to solar energy transport were described effectively with geometrical optics, accounting for macroscale energy modeling by vector propagation. For successive ray propagation and transformation, using Fermat’s principle to derive instances of the Snell-Descartes Law, the following refraction and transfer equations are used in Eq.2.3a-2.3b (referred to as the ‘YNU’ raytracing method) [54]:

\[ n'u' = nu - y\phi \]  
\[ y' = y + n'u'(t'/n') \]  

where \( t \) is the extent of ray transfer, \( t'/n' \) is the reduced distance for optical path length dependent on a specific refractive index \( n \), \( u \) and \( y \) are the respective deviated angle and height from the direction of propagation, and \( \phi \) is the change in optical power based on a ray interaction with a new surface.

Perturbations to an average wavelength taken at a specific refractive index \( n \), incorporate spectral absorptance considerations that are proportional to material density. Spectral-dependence in terms of variation of energy propagation, based on Cauchy descriptors of dispersion, is described according to the Sellmeier equation in Eq.2.4:

\[ n^2(\lambda) = A + \sum_i \frac{B_i\lambda^2}{\lambda^2 - C_i} \]  

where \( A_i, B_i, \) and \( C_i \) are constants, Additionally, wavelength has a temperature
dependence (Eq.2.5) according to material expansion coefficient $\alpha$ and thermo-optic coefficient $\zeta$:

$$\frac{\delta \lambda}{\lambda} = (\alpha + \zeta) \Delta T$$  \hspace{1cm} (2.5)

An extension to the Sellmeier equation can describe wavelength-dependent refractive index $n(\lambda)$ containing temperature-dependent modification $n(\lambda, T)$, in Eq.2.6:

$$\frac{dn(\lambda, T)}{dT} - n(\lambda, T) = -\sqrt{A + \sum_i \frac{B_i \lambda^2}{\lambda^2 - C_i}}$$  \hspace{1cm} (2.6)

**Electromagnetic Theory**  Energy transport, as governed by the ray theory of light, was later refined in the 19th century to account for interference, diffraction, and coherent effects based on relational dependencies of propagating optical fields, giving rise to the wave theory of light, wherein, electric and magnetic charges collectively produce a field, which in turn produce a force; the Lorentz Force Law describes the generation of light:

$$\vec{F} = q\vec{E} + q\vec{v} \times \vec{B} = q(\vec{E} + \vec{v} \times \vec{B})$$  \hspace{1cm} (2.7)

Of the fields of importance, the Electric Field $\vec{E}$ (sometimes referred to as the macroscopic field) and the electric displacement field $\vec{D}$ (microscopic field) are synonymous in that the latter has a polarization dependence based on the constitutive relations (it’s generally not completely dependent on length scales, despite the nomen-
clature. Additionally, the magnetic field $\mathbf{H}$ and magnetic induction $\mathbf{B}$ is comparable in that the latter field has dependence on magnetic induction or spin magnetic moments). The extent of polarization $\mathbf{P}$ and magnetization $\mathbf{M}$ are defined from the constitutive relations that introduce the electric displacement $\mathbf{D}$ and magnetic induction $\mathbf{B}$) as:

$$
\mathbf{D}(\mathbf{r}, t) = \varepsilon \mathbf{E}(\mathbf{r}, t) + \mathbf{P}(\mathbf{r}, t) \tag{2.8a}
$$

$$
\mathbf{B}(\mathbf{r}, t) = \mu \mathbf{H}(\mathbf{r}, t) + \mathbf{M}(\mathbf{r}, t) \tag{2.8b}
$$

where $\varepsilon_0 = 8.854 \times 10^{-12} \text{ F m}^{-1}$ is the permittivity of free space, $\mu_0 = 4\pi \times 10^{-7} \text{ H m}^{-1}$ is the permeability of free space. The constitutive relations show that while the electric field types $\mathbf{E}$ and $\mathbf{D}$, as well as the magnetic field types $\mathbf{H}$ and $\mathbf{B}$, are related, they differ based on units and their physical descriptors. Comparably, the magnetic susceptibility $\chi_m$ is: $\mathbf{M} = \chi_m \mathbf{H}$ and $\mathbf{B} = \mu_0 (1 + \chi_m) \mathbf{H}$.

This representation of Maxwell’s Equations are linear, shift-invariant (LSI), and as such, polarization is related directly to electric field based on the electric susceptibility $\chi_e$, where $P(\mathbf{r}, t) = \varepsilon_0 \chi_e \mathbf{E}(\mathbf{r}, t)$. While this ignores nonlinear effects (Pockels effect, Kerr effect, Franz-Keldysh effect, etc.), the mathematical treatment herein will assume these factors to be negligible [55]. Light must have both $\mathbf{E}$ and $\mathbf{H}$, and a non-zero Poynting vector $\mathbf{S}$, where $\mathbf{S} = \mathbf{E} \times \mathbf{H} \neq 0$. 
Maxwell’s Equations in macroscopic form are [56]:

\[ \nabla \cdot \vec{D}(\vec{r}, t) = \rho_{\text{free}} \quad (2.9a) \]
\[ \nabla \times \vec{H}(\vec{r}, t) = \vec{J}_{\text{free}} + \frac{\partial \vec{D}(\vec{r}, t)}{\partial t} \quad (2.9b) \]
\[ \nabla \times \vec{E}(\vec{r}, t) = -\frac{\partial \vec{B}(\vec{r}, t)}{\partial t} \quad (2.9c) \]
\[ \nabla \cdot \vec{B}(\vec{r}, t) = 0 \quad (2.9d) \]

where \( \rho_{\text{free}} \) [C m\(^{-3}\)] is the free charge density, \( \vec{J}_{\text{free}} \) [A m\(^{-2}\)] is the free current density, \( \vec{P}(\vec{r}, t) \) [C m\(^{-2}\)] is the polarization, and \( \vec{M}(\vec{r}, t) \) [Wb m\(^{-2}\)] is the magnetization. The gradient (when acting on a scalar field: \( \nabla \Psi = \left[ \frac{\partial \Psi}{\partial x}, \frac{\partial \Psi}{\partial y}, \frac{\partial \Psi}{\partial z} \right] \)), divergence (produces a scalar field: \( \nabla \cdot \vec{\Psi} = \frac{\partial \Psi}{\partial x} + \frac{\partial \Psi}{\partial y} + \frac{\partial \Psi}{\partial z} \)), and curl (\( \nabla \times \vec{\Psi} \), producing a vector field) are defined for Cartesian coordinates, and charge and current densities are related by the continuity equation:

\[ \nabla \cdot \vec{J}(\vec{r}, t) + \frac{\partial \rho_{\text{free}}(\vec{r}, t)}{\partial t} = 0 \quad (2.10) \]

Equation 2.9a describes Gauss’ law, originally \( \nabla \times \vec{E}(\vec{r}, t) = 0 \), indicating that in the absence of any free charges, the net divergence of the electric field is nullified. For propagation through a medium free of bound charges and current enclosed within the propagation volume, as well as one free of polarization and magnetization events:
\begin{align}
\n\nabla \cdot \vec{D}(\vec{r}, t) &= \mu_{\text{free}} = 0 \quad \text{(2.11a)} \\
\n\nabla \times \vec{H}(\vec{r}, t) &= \frac{0}{\sigma_{\text{free}}} + \frac{\partial \vec{D}(\vec{r}, t)}{\partial t} = \frac{\partial \vec{D}(\vec{r}, t)}{\partial t} \quad \text{(2.11b)} \\
\n\vec{D}(\vec{r}, t) &= \epsilon \vec{E}(\vec{r}, t) + \vec{B}(\vec{r}, t) = \epsilon \vec{E}(\vec{r}, t) \quad \text{(2.11c)} \\
\n\vec{B}(\vec{r}, t) &= \mu \vec{H}(\vec{r}, t) + \vec{M}(\vec{r}, t) = \mu \vec{H}(\vec{r}, t) \quad \text{(2.11d)} \\
\n\n\end{align}

Taking the curl of (Eq.2.9b) and (Eq.2.9c) produces the following:

\begin{align}
\n\n\nabla \times (\nabla \times \vec{E}(\vec{r}, t)) &= -\mu_0 \epsilon_0 \frac{\partial^2 \vec{E}(\vec{r}, t)}{\partial t^2} \quad \text{(2.12a)} \\
\n\n\nabla \times (\nabla \times \vec{H}(\vec{r}, t)) &= -\mu_0 \epsilon_0 \frac{\partial^2 \vec{H}(\vec{r}, t)}{\partial t^2} \quad \text{(2.12b)} \\
\n\n\n\end{align}

By this formulation, for light propagation in an isotropic material medium, simplifications are permitted based on the substitutions referenced above to a medium void of free charges and free currents, where \( \rho_{\text{free}} = 0 \) and \( \vec{J}_{\text{free}} = 0 \). Based on the vector identity \( \nabla \times (\nabla \times \vec{A}) = \nabla (\nabla \cdot \vec{A}) - \nabla^2 \vec{A} \), the wave equation can be formulated for electric and magnetic fields to obtain the wave equation (Eq.2.13) and solution of a plane wave (Eq.2.14):

\begin{align}
\n\n\nabla^2 \Psi(\vec{r}, t) &= \mu \epsilon \frac{\partial^2 \Psi(\vec{r}, t)}{\partial t} \quad \text{(2.13)} \\
\n\n\Psi(\vec{r}, t) &= \Psi_0 \exp [j(\omega t - \vec{k} \cdot \vec{r} + \phi_0)] \quad \text{(2.14)} \\
\n\n\end{align}
where \( \vec{r} \) denotes the propagation direction, \( t \) the temporal evolution, \( \omega \) is the angular frequency of the light wave, \( \vec{k} \) is the propagation vector containing the wavenumber (where \( k = k(\lambda) \), \( \phi_0 \) is the phase-shift of the wave, \( \Psi_0 \) is the wavefunction amplitude, and \( j \) is the imaginary unit \( \sqrt{-1} \). The time-evolution of the output signal is characterized by successive discretized propagation and transformation events based on statistical interpretations of the underlying material transport considerations.

The solve the differential wave equation (Eq.2.13) and obtain the plane-wave solution in (Eq.2.14), formulation using Green’s functions is employed to interpret fundamental solutions using approximations from scalar diffraction theory.

Maxwell’s equations can be used to interpret emanating fields \( (\vec{\Psi}) \) from known sources \( (\Psi) \), all of which use linear operators \( \mathcal{L} \) to posit:

\[
\Psi = \mathcal{L}\vec{\Psi}
\quad (2.15)
\]

where the inverse transfer function \( \mathcal{L}^{-1} \) is an inverse linear operator based on Green’s Functions, invoked to solve for the fields present:

\[
\vec{\Psi} = \mathcal{L}^{-1}\Psi = \mathcal{L}^{-1}\mathcal{L}\vec{\Psi}
\quad (2.16)
\]

The associated Green’s Function \( G(t, t_0) \) is based on the type of inverse operator acting on the light source, where \( dG(t, t_0) = \delta(t - t_0) \), and the source term is expressed as:
\[ \Psi = \int_{-\infty}^{\infty} G(t, t_0) \tilde{\Psi}(t_0) dt_0 \]  

(2.17)

and

\[ \frac{\partial \Psi(t)}{\partial t} = \frac{\partial}{\partial t} \int_{-\infty}^{\infty} G(t, t_0) \tilde{\Psi}(t_0) dt_0 = \int_{-\infty}^{\infty} \tilde{\Psi}(t_0) \delta(t - t_0) dt_0 = \tilde{\Psi}(t) \]  

(2.18)

The formulation of a superposition of linear operators, taken as the impulse response of an inhomogenous linear differential equation, allows for obtaining the solution set of emanating fields given known boundary conditions for where the sources are not defined.

The simplified, revisited geometrical optics formulation allows for the interpretation of a plane wave propagating toward a boundary at an angle \( \theta \), where reflection occurs from the boundary normal \( \hat{n} \) at angle \( \theta' \), and refraction occurs through the boundary surface at angle \( \phi \) from the opposing boundary normal \(-\hat{n}\), the relation \( \vec{k} \cdot \vec{r} = \vec{k'} \cdot \vec{r'} = \vec{k''} \cdot \vec{r''} \) (referencing propagation vectors for incident ray, reflected ray, refracted ray, respectively) must occur at the boundary, where colocated vector references must be equal. For unitary vectors \( \vec{r} \), the projection on the surface boundary (not the normal to the boundary) is given as \( \vec{k} \sin \theta = \vec{k'} \sin \theta' = \vec{k''} \sin \phi \).

For reflection, the propagation vector is equal \( \vec{k} = \vec{k'} \), so we conclude that \( \sin \theta = \sin \theta' \), or as stated by the Law of Reflection:
\[ \theta = \theta' \quad (2.19) \]

For refraction, the ratio of the transmitted wave to the incident wave can be equated as follows:

\[
\frac{k'}{k} = \frac{w'/u'}{w/u} = \frac{c/u'}{c/u} = \frac{n_2}{n_1} = \frac{\sin \theta}{\sin \phi} \quad (2.20)
\]

where \( u \) is the phase velocity with index of refraction definition \( n = c/u \), the ratio of light speed in a vacuum to light speed in a material medium. In this case, common notation has \( \theta = \theta_1 \) and \( \phi = \theta_2 \) to compare to the associated media refractive index. The Snell-Descartes Law can therefore be written as:

\[
\boxed{n_1 \sin \theta_1 = n_2 \sin \theta_2} \quad (2.21)
\]

The ray optics formulation, governing the laws of reflection and refraction, gives a descriptor for the Monte Carlo raytracing established in Study.2, where a vector set of equations in describing propagating ray paths, and reflection events based on deviation from interacting with surface normals is given in Eq.2.22.

\[
r_{k+1} = r_k + t_k s_k \quad (2.22a)
\]

\[
s_{k+1} = \kappa s_k + \chi n_{k+1} \quad (2.22b)
\]

where \( \kappa = n_k/n_{k+1} \) is the ratio of subsequent refractive indices and \( \chi = -(s_k \cdot \)
\( n_{k+1}(\kappa - \sqrt{\kappa^2 + (1 - \kappa^2)/(s_k \cdot n_{k+1})^2}) \), describing the optical power at an interface.

**Quantum Theory** Light, comparably described in quantum mechanics formulations, can be modeled according to the Schrödinger Equation in Eq. 2.23.

\[
\hat{H} |\Psi(\vec{r}, t)\rangle = \hat{E} |\Psi(\vec{r}, t)\rangle \tag{2.23}
\]

where the kinetic energy operator \( \hat{T} = -\frac{\hbar^2}{2m} \nabla^2 \) and potential energy operator \( V(\vec{r}) \) combine to give total energy (or the energy operator \( \hat{E} = j\hbar \frac{\partial}{\partial t} \) if featuring temporal dependence) equal to the Hamiltonian operator \( \hat{H} = \hat{T} + \hat{V} \).

The total description of the time-dependent Schrödinger Equation is given in Eq. 2.24:

\[
-\frac{\hbar^2}{2m} \nabla^2 \Psi(\vec{r}, t) + V(\vec{r})\Psi(\vec{r}, t) = j\hbar \frac{\partial \Psi(\vec{r}, t)}{\partial t} \tag{2.24}
\]

This formulation is useful in describing effects of photon interaction with the solar absorber coating material. Essentially, specific types of scattering conditions are more useful with a quantum mechanics descriptor. The essence of Wave-Particle duality is an important consequence for light interference (multiple ‘waves’ of light superimposing to increase or decrease wave amplitude for filtering effects), diffraction (a ‘bending’ of light succeeding contact with aperture edges for solar irradiation effects on absorber material topology), polarization (spectrally-dependent effects of solar irradiation based on varying orientations of EM fields modified after transmission...
through the earth’s atmosphere), and scattering effects (changes in photon momentum when interacting with a surface). The quantum description, in particular, is useful for describing the effects of photon absorption, where electrons present in energy orbitals are excited to higher energy states due to incident solar irradiation. In such a temporary excited state, atoms that absorb photons will either undergo emission (re-emit energy in different spectral forms) or release energy in the form of heat – special interest in Study.1 is identifying low emittance materials for nullifying ‘waste heat’, supporting energy conversion efficiency improvements for CSP solar receivers.

Heat Transfer

Heat transfer phenomenology is typically modeled according to varying rates of diffusion, where phonon transport occurs in solids (conduction), fluids (convection/advection), and internal energy transfer from electromagnetic waves (radiation). The heat (or diffusion) equation, is given in Eq. 2.25:

$$\frac{\partial \Psi}{\partial t} = \gamma \nabla^2 \Psi(\vec{r}, t)$$

(2.25)

where $\gamma$ is a diffusivity constant (in this instance, thermal diffusivity) – for large values of $\gamma$, the diffusion rate is faster, as concavity (the 2nd derivative) is proportional to temporal changes.

Heat flux in the infrared regions was modeled according to the Wien Approximation and Rayleigh-Jones Law, leading to the Ultraviolet Catastrophe. Fabricating
such solar absorber coatings in this work relies on an optimization for low-cost synthesis while allowing high-temperature operation and spectrally-selective behavior; fine-tuning irradiance for spectral dependence and temperature is determined by the Planck Radiation Law (Eq. 2.26, correcting the ultraviolet catastrophe), where $B$ is the spectral- and temperature-dependent radiance, and $\lambda$ is the spectral wavelength:

$$B(\lambda, T) = \frac{8\pi hc}{\lambda^5} \frac{1}{\exp(hc/\lambda kT) - 1}$$  \hspace{1cm} (2.26)$$

$$B(\nu, T) = \frac{8\pi h\nu^3}{c^3} \frac{1}{\exp(h\nu/kT) - 1}$$  \hspace{1cm} (2.27)$$

Solar Energy Transport

Solar spectral regions defined in this study are taken from the ASTM-G173 spectral profile [7], describing irradiated solar energy upon the earth. In accounting for atmospheric absorption and scattering effects, such solar irradiation profiles are cited for different geographic regions of the earth, as well as extra-terrestrially from relative measurements of satellite. Typically this value is approximated as 1.3530 kW m$^{-2}$, 0.970 kW m$^{-2}$, and 0.930 kW m$^{-2}$ for extra-terrestrial, direct desert sea level, and direct standard sea level respectively. These values change based on a host of factors, such as measurement timescale, properties of the solar detector making the measurement, and annual variation due to planetary orbital eccentricity. Meinel et. al [9] describes an equation that fits the data to calculate solar flux intensity $I(z)$ in units
kW m$^{-2}$:

$$I(z) = I_0 \exp[-\alpha(\sec(z))^{\beta}]$$  \hspace{1cm} (2.28)

where $I_0$ is the exoatmospheric solar flux, $z$ is the zenith distance, and $\alpha, \beta$ are two numerical constants: $\alpha = 0.357, \beta = 0.678$. This equation is refined with solar flux intensity data fitted at different elevations:

$$I(z, h) = I_0(1 - \gamma h) \exp[-\alpha(\sec(z))^{\beta}] + \gamma h I_0$$  \hspace{1cm} (2.29)

where $h$ is the elevation in kilometers, $\gamma$ is an altitude scaling term equal to $\gamma = 0.14$ km$^{-1}$, and the overall equation is applicable to a few kilometers of tolerance around sea level, since the linear term $(1-\gamma h)$ only approximates solar flux intensity to the confined tolerance $(1-\gamma h) < 1$. Fig.2.1 features the ASTM-G173 solar irradiation profile, juxtaposed with a blackbody radiation at $T = 750$ °C based on Eq.2.26. The optimal efficiency metric based on the ratio of absorbed energy to total energy (and accounting for thermal losses) is described by Eq.2.32.
In modeling material transport phenomena at mesoscale, there exist many custom approaches depending on application. Computational codes are generally compatible with either atomic scale simulations (electronic structure theory, density functional theory, molecular dynamics, etc.) [57]. Macroscopic upscaling to finite-difference and finite-element methods [58] are ubiquitous in computer-aided design software platforms. Solar energy, represented as propagating electromagnetic waves, takes into account scattering and dispersion effects to track light-matter interaction phenomenology, which require models based on EM interaction at mesoscale length scales. For optical scattering, photons (packets of electromagnetic energy) interact with a charged particle. Dispersion accounts for the spectral-dependent deviation of light based on
the energy (and frequency) of incident photons interacting with bound charges in a material. Most radiative heat transfer can be modeled through raytracing calculations, given that such problems are largely deterministic.

For ideal solar thermal conversion, solar radiant energy is converted to a heat-transfer receiver in the CSP process while minimizing any associated losses. Solar Irradiance Spectra, on a national average in the United States, can be approximated by a Gaussian profile with functional form:

\[
\phi(\lambda, T) = c_1 \lambda^{-5} \exp \left( \frac{c_2}{\lambda \tau} \right)
\]

where \( \tau \) is the surface temperature of a solar absorber, and \( c_1, c_2 \) are coefficients equal to \( c_1 = 3.7814 \times 10^{-16} \text{ W m}^{-2} \) and \( c_2 = 1.4388 \times 10^{-2} \text{ m K} \).

Specifically to the 21st-century energy portfolio, classifying energy transfer through spectrally-dependent solar irradiation is performed with Eq.2.31, describing an efficiency metric of absorbed energy per total energy:

\[
\eta_{\text{abs}}(\lambda, T, t) = \frac{\int_0^\infty \int_0^\infty (1 - R(\lambda, t)) E(\lambda) dt d\lambda - \frac{1}{c} \left( \int_0^\infty \int_0^\infty (1 - R(\lambda, t)) B(\lambda, T) dt d\lambda \right)}{\int_0^\infty E(\lambda) d\lambda}
\]

(2.31)

or without temporal dependence as:
\[
\eta_{\text{FoM}}(\lambda, T) = \frac{\int_{\lambda_1}^{\lambda_2} (1 - R(\lambda)) E(\lambda) d\lambda - \frac{1}{C} \left( \int_{\lambda_1}^{\lambda_2} (1 - R(\lambda)) B(\lambda, T) d\lambda \right)}{\int_{\lambda_1}^{\lambda_2} E(\lambda) d\lambda}
\]  

(2.32)

where the figure-of-merit efficiency \( \eta_{\text{FoM}} \) includes functional dependence on the boundaries of integration \( \lambda_1 \) and \( \lambda_2 \), \( R(\lambda) \) is the temporal-averaged spectral reflectivity, \( E(\lambda) \) is the time-averaged incident spectral irradiance, \( B(\lambda, T) \) is the Planck function at temperature \( T \), and \( C \) is the solar concentration factor [59].

Spectrally-selective solar absorber coatings can absorb ultraviolet, visible, and near-infrared solar irradiation while limiting spontaneous thermal radiation from emittance at higher wavelengths, as predicted by behavior from the Stefan-Boltzmann Law (Eq.2.33) as a simplification of Eq.2.31 assuming perfect absorptance, where \( M \) is the radiant exitance, \( k \) is the Boltzmann constant, \( h \) is Planck’s constant, \( c \) is the speed of light in vacuum, \( \sigma \) is the Stefan-Boltzmann constant, and \( T \) is the thermodynamic temperature:

\[
M(T) = \frac{2\pi^5 k^4}{15c^2h^3} T^4 = \sigma T^4
\]

(2.33)

where \( M(T) \) is the exitance defined in Table.2.2.

Determining optimal concentration factors based on operating temperatures requires taking into consideration the Carnot efficiency \( \eta_{\text{Carnot}} \) in a combined efficiency
equation described in Eq. 2.34.

$$\eta(T, C') \approx \eta_{\text{FoM}}\eta_{\text{Carnot}} = \left(1 - \frac{\sigma T^4}{CE(\lambda)}\right) \left(1 - \frac{T_C}{T}\right) \quad (2.34)$$

where the gradient $C_{\eta_{\text{max}}}$ is defined in Eq. 2.35 and depicted as the black dashed line in Fig. ??.

$$C_{\eta_{\text{max}}} = \frac{\sigma T^4(4T - 3T_C)}{E(\lambda)T_C} \quad (2.35)$$

Figure 2.2: Operating efficiency of concentrated solar power systems with recommended concentration factor based on the absorber coating’s reliable operating temperature
The above formalism in Eq.2.31-2.33 influences the recommended CSP operating conditions given by Fig. 2.2, suggesting temperature $T$ and concentration factor $C$ for best-case operating efficiencies.

**Diffraction & Optical Scattering**

Upon particle interaction, EM fields are subject to optical scattering events, which propagate electromagnetic energy into a variety of directions based on specular and diffuse reflectance and absorptance criteria (transmittance is effectively nullified based on the configuration of the material). Precisely calculating scatter contributions correlate directly to the material’s absorptance capacity, as determining scatter can allow for simulating the extent of reflectivity and absorptivity [60, 61]. For purposes of simulation, the initial wave field comprises fundamental mode Gaussian beamlets in the form of:

$$u_0(x, y) = \exp \left( -\pi \left( \frac{x - x_0}{a_0} \right)^2 \right) \exp \left( -\pi \left( \frac{y - y_0}{b_0} \right)^2 \right) = \text{gaus} \left( \frac{x - x_0}{a_0} \right) \text{gaus} \left( \frac{y - y_0}{b_0} \right)$$

(2.36)

where $a_0$ and $b_0$ are the Gaussian beamlet cross-sectional widths, superimposed to create an irradiation profile with successive incident energy and propagated energy, either absorbed in the material or reradiated through infrared emission. In diagnosing optical transport through determining experimental solar absorptive response using a laboratory setup consisting of a tungsten-halogen light source, diffraction effects are
noted based on the irradiated area of the sample under test, described by the Fresnel transform in Eq.2.37.

\[ u_2(x_2, y_2) = \frac{e^{jk_{z_{12}}}}{j\lambda_{z_{12}}} e^{\frac{\pi}{\lambda_{z_{12}}}(x_2^2 + y_2^2)} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u_1(x_1, y_1) e^{\frac{\pi}{\lambda_{z_{12}}}(x_1^2 + y_1^2)} e^{-j\frac{2\pi}{\lambda_{z_{12}}}(x_1 x_2 + y_1 y_2)} dx_1 dy_1 \]

\[ = \frac{e^{jk_{z_{12}}}}{j\lambda_{z_{12}}} e^{\frac{\pi}{\lambda_{z_{12}}}(x_2^2 + y_2^2)} \mathcal{F}\{u_1(x_1, y_1) e^{\frac{\pi}{\lambda_{z_{12}}}(x_1^2 + y_1^2)}\}|_{\xi = \frac{x_2}{\lambda_{z_{12}}}, \eta = \frac{y_2}{\lambda_{z_{12}}}} \]

\[ = \frac{e^{jk_{z_{12}}}}{j\lambda_{z_{12}}} e^{\frac{\pi}{\lambda_{z_{12}}} r^2} \mathcal{F}\{u_1(r_1) e^{\frac{\pi}{\lambda_{z_{12}}} r^2}\}|_{\rho = \frac{r_2}{\lambda_{z_{12}}}} \]

(2.37)

where \( u(x, y) \) is the optical field, \( r \) is the source and or detector radius, and \( z_{12} \) is the distance spanning the source and the detector. In Fig. 2.3, the CSP absorber coating sample surface is irradiated with diffracted light inside of an integrating sphere (left), spectrally narrowly-windowed from a scan monochromator slit and aperture (center); thin paper placed at the sample plane presents the diffracted irradiance profile more clearly (right).
Photon-matter interactions at interfaces where reflection events are noted, will tend to introduce angular-dependent deviations according to scattering. Elastic scattering (Rayleigh scattering), ideally occurs with no photonic energy loss ($\Delta E = 0$) when a photon $h\nu$ is sent into a material, it excites an electron to a higher energy state, causing it to cascade back down to the original energy state, and it omits a photon with the same energy as the input photon. The other case of inelastic scattering introduces Raman scattering, where $\Delta E \neq 0$, and there is no emitted photon since the electron that is excited by the incident photon does not return to the same energy level at which it originally started. Regardless of the scattering type, the intensity of the photon is defined by:

$$I = I_0 \frac{1 + \cos^2 \theta}{2R^2} \left( \frac{2\pi}{\lambda} \right)^4 \left( \frac{D}{2} \right)^6 \left( \frac{n^2 - 1}{n^2 + 2} \right)^2 \propto \lambda^{-4}$$  \hspace{1cm} (2.38)$$

where $\theta$ is the scattering angle, $R$ is the source-particle distance, $I_0$ and $I$ are the
initial and final intensity profiles of light, $D$ is the average particle diameter, and $\lambda$ is the wavelength of light. The $I \propto \lambda^{-4}$ term essentially governs the larger scattering at shorter wavelengths.

An important consequence of scattering is that the shorter the wavelength of light, the more scattering that occurs (hence, why the sky is blue: more scattering events for high frequency, low wavelength photons).

Statistical Summary

Correlation between image processing data, scatter functions, and experimentally-derived irradiance calculations are developed to accurately define simulation conditions and make testable predictions about refining the coating syntheses. Accuracy metrics are determined in comparing other known simulation methods and calibration data to confirm that proposed scatter functions match experimental output from optical simulation data. Most data can be fit to the general normal or Gaussian Distribution, especially given the Law of Large Numbers, where for many events, all distributions tend to the normal distribution given in Eq.2.39.

$$f(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2 \pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$  \hspace{1cm} (2.39)

Generally, particle sizes are described according to a work by Waloddi Weibull [62], where the Weibull continuous probability distribution is defined in Eq.2.40, defining the generalized Weibull distribution probability density:
\[
f(x; \theta, k) = \begin{cases} 
\frac{k}{\theta} \left(\frac{x}{\theta}\right)^{k-1} \exp\left(-\left(\frac{x}{\theta}\right)^{k}\right), & x \geq 0 \\
0, & x < 0
\end{cases} 
\]  
\tag{2.40}

where \( k \) is the shape parameter and \( \theta \) is the scale parameter. More generally, fit or experimental data can be made by the Gamma distribution given in Eq.2.41, as the Weibull distribution is a special case of the Gamma distribution.

\[
f(x; k, \theta) = \frac{x^{k-1} \exp(-x/\theta)}{\theta^k \Gamma(k)}, \quad x > 0 
\tag{2.41}
\]

where \( k \) is the shape parameter and \( \theta \) is the scale parameter. This distribution is used for studies in tracking algorithmic comparisons for tracking particle and pore size distribution of the solar absorber materials.

Particle statistics are governed entirely by conditions for image sampling on morphological data obtained through microscopy, based on number of pixels available on a detector area. The Nyquist-Shannon Sampling Theorem (Eq.2.42) bridges phenomenology of continuous-time (analog) signals and discrete-time (digital) signals, describing a sufficient condition to determine a spatial analysis area that allows for a discrete sequence of samples to capture the entirety of the information from a continuous-spatial signal of finite bandwidth and prevent aliasing during signal reconstruction.

\[
f(x) = \sum_{n=-\infty}^{\infty} f\left(\frac{n}{2B}\right) \text{sinc}\left(2B(x - \frac{n}{2B})\right) 
\tag{2.42}
\]
while under certain ‘slowness’ conditions for \( f(x) \), a discrete-spatial sequence can be resampled to its original signal by:

\[
    f(x) = \sum_{n=-\infty}^{\infty} f[n] \text{sinc}\left(\frac{x - nX_s}{X_s}\right)
\] (2.43)

Multiresolution Analysis

Using multiresolution analysis, an absorptance optimization criterion is based on ideal coating morphology suggested – synthesis refinement is performed to experimentally compare successive results. A signal, representing the spatial evolution of some material variable can be represented as a linear decomposition of basis functions in \( \mathbb{R}^N \):

\[
    \tilde{f}(\tilde{r}) = \sum_{n=1}^{N} \alpha_n \tilde{u}_n(\tilde{r}_n)
\] (2.44)

where \( \alpha \) represents scalar coefficients affecting the amplitude of the represented functional set of \( \tilde{u} \). Popular basis functions for \( u_n \) include the Taylor series \( u = t^k \) and the Fourier series \( u = \exp(-j2\pi kx) \). The general Fourier Series is defined in Eq.2.45 as:

\[
    f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left( a_n \cos 2\pi kt + b_n \sin 2\pi kt \right)
\] (2.45)

which can be summed to give the Fourier Transform as defined in Eq.2.46, used for interpreting spatial frequencies of morphological data, establishing a resolution
criterion for SEM images that give accurate data.

\[ F(\xi) = \sqrt{\frac{|b|}{(2\pi)^{1-a}}} \int_{-\infty}^{\infty} f(x) \exp(-jb\xi x) dx \]  \hspace{1cm} (2.46)

where \{a, b\} represent the Fourier parameters: \{0,1\} for physics, \{-1,1\} for data analytics, \{1,-1\} for signal processing notations, and \(f(x)\) is represented by a weighted sum of sinusoids, given by the complex exponential \(\exp(-jb\xi x)\) so that each frequency has a single complex coefficient. Examples of Fourier Transform pairs are given in Fig.2.4, where a continuous Gaussian function reproduces another scaled Gaussian (column 1), spaced Gaussians interpreted as the convolution of a the original function with a Dirac comb produce a sampled Gaussian (column 2), a sampled Gaussian function at specific impulses produce a superposition of continuous sinusoids (column 3), and sampling the spaced functions from column two produces another comparable scaled function (column 4).

![Figure 2.4: Visualization of fourier transform pairs, comparing continuous-time (columns 1,2) and discrete-time (columns 3,4) signals](image-url)
Signals (spatial or temporal representations) localized entirely within the frequency domain contain information that is not observed in the reciprocal domain unless an inverse Fourier transform operation is applied. While the entirety of the signal is recoverable, this does not allow for dual measurements – this is akin to the Heisenberg uncertainty principle in Eq.2.47a-2.47b:

\[ \sigma_x \sigma_p \geq \frac{\hbar}{2} \]  

\[ \sigma_t \sigma_\omega \geq k \]  

In Eq.2.47b, \( k = p/\hbar \), where \( k \) is the wavenumber, inversely proportional to the reduced Planck constant \( \hbar = h/2\pi \). The convolution theorem is defined in Eq.2.48, where mathematical convolution is equal to the product in reciprocal space; this is useful for the material analysis methods used in Study.1, as well as for calculation summaries of frequency data used in Study.2. It is also related to statistical correlation, where the functional autocorrelation is performed on optical microscopy data, to determine statistics about localizing microscopy data for approximating a scatter function.
Convolution Theorem:

\[ \mathcal{F}(g(x)) = \mathcal{F}(f(x) * h(x)) \]
\[ = \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} f(\alpha) h(x - \alpha) d\alpha \right) \exp(-j2\pi \xi x) dx \]
\[ = \int_{-\infty}^{\infty} f(\alpha) H(\xi) \exp(-j2\pi \xi \alpha) d\alpha \]
\[ = F(\xi)H(\xi) \]

(2.48)

\[ x(t) * h(t) = y(t) \]
\[ X(f) H(f) = Y(f) \]

Given an array of real-valued inputs, the Discrete Fourier Transform (DFT) will output complex numbers, as it gives frequency content as both magnitude and phase information. The magnitude relates to the energy of the sinusoidal waves used to construct a signal or an image, while the phase correlates to the position of the sinusoids in the spatial/temporal domain. It is noted that the spatial frequencies are not described in terms of real and imaginary parts exclusively. Implementation of Fourier transform methods on sample morphology data uses the Fast Fourier Transform (FFT), an important algorithm for computing the DFT, reducing computation time from \( n^2 \) to \( n \log(n) \) complexity.
Lastly, for determining detector errors from microscopy intensity datasets, the Wavelet Transform can be implemented as a filtering technique, based on applying small spatial- and frequency-localized band-limited waveforms (wavelets) obtained from scaling and translation to different representations of basis functions [63]:

\[
\Psi(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(x) \psi^*\left(\frac{x - b}{a}\right) dx \propto \int_{-\infty}^{\infty} f(x) \psi^*_{a,b}(x) dx
\]  \hspace{1cm} (2.49)

where \( \psi_{a,b}(x) \) is the analyzing wavelet. A function of a single variable has an output of two variables in scale and translation, allowing for the calculation of different space/time and frequency resolution (Fig.2.5).

Figure 2.5: Visualization of wavelet transform methods, leveraging position-frequency or time-frequency content according to varying scale parameters, reproduced from Mallat et. al. [8]

A wavelet expansion adds a second index to Eq.2.44, where a spatial-frequency localization can be determined over a finite interval in regards to scaling and translation

40
parameters (rather than pure frequencies from the Fourier analogue):

\[ g(t) = \sum_{k=-\infty}^{\infty} c(n)\varphi_n(t) + \sum_{m=0}^{\infty} \sum_{n=-\infty}^{\infty} d(m,n)\psi_{m,n}(t) \]  

(2.50)

Wavelet transforms have been prominent for data compression, though the flexibility of wavelets allows microscopy images of two spatial variables to generate output of varying frequency resolution. It can be used to infer particle statistics more precisely in the frequency domain compared to other transformation methods that are applied in the spatial domain based on convolution of image filters [63]. Wavelets can help to track particle shape and sizing [64, 65], texture analysis [66, 67], particle location [68], feature extraction [69, 70, 71], and noise removal [72]. They are applicable to infer particle statistics compared to other possible testable methods in the spatial domain, such as roughness, location, layout/density, de-noising and resolution, particle sizing/distribution, and contours/feature extraction.

Studies will conclude by introducing comparisons against other methods to determine best-case and relative accuracy for specific microscopy applications, as particle scatter is calculated for extracting topological information to propose energy transport detail. Special scatter measurements will be necessary to ensure the accuracy of the simulation routines devised to support experimental studies. Dissertation experimental component may be useful in designing a setup at UNLV that can measure samples at elevated temperatures. The extent of the optical transport theory developed in this section will be used subsequently for three established studies: developing
an experimental framework for scalable manufacturability of solar absorber coating materials and implementing statistical analyses from simulation methods to discern fabricated coating quality. Developed models are implemented, with the ambition to improve material properties based on predictions of morphological behavior.
CHAPTER 3

EXPERIMENTAL SYNTHESES

“Come celebrate with me that everyday something has tried to kill me and has failed.”  
- Lucille Clifton

This preliminary study demonstrates the utility of intrinsic absorber coatings based on surface-textured performance enhancements [34]. Fabricated cuprous oxide-based nanoparticles, when ported to a viscous slurry solution via a silicon dioxide (SiO$_2$) binder, were optionally treated with sacrificial polymer beads (SPB) made of poly(methyl-methacrylate) of diameter $D_{\text{SPB}} = 1$–$5 \mu$m to explore possibilities for effectively surface-texturing different light-trapping geometries [31]. For cost-effective upscaling so as to meet the needs of near-term CSP production competitiveness, process optimizations were implemented for the reduction of waste byproducts, low-cost of process reactants, and ease of synthesis while producing high-purity materials [73].

Candidate Materials for Synthesis

Initially, a collection of inorganic oxide materials were tabulated as candidates for spectrally-selective absorber coatings, as would be useful for concentrated solar power systems and concentrated photovoltaics [4]. Primarily, this is due to their air-
stable nature given oxide-containing constituents, configuration and maintainability at high-temperatures based on metallic properties, and crystallographic arrangement that absorbs energy in the UV-VIS spectral range and rejects it in the mid-to-far infrared bands. The target studies in this work are to synthesize spectrally-selective coatings survivable at $T = 700^\circ$C or higher working temperatures [74].

A collection of different nanomaterial syntheses have been proposed for fabricating solar absorber coating materials that intrinsically exhibit spectrally-selective behavior [28]. The production of solar absorber coatings are commonly used by varying methods of synthesis and deposition, such as sputtering [75, 76], spray pyrolysis [77, 78, 79, 80], electrodeposition [81, 82, 83, 84], sol-gel methods [85, 86, 87, 88, 89], hydrothermal syntheses [90, 91, 92, 93], and high-energy ball milling (HEBM) [94], etc.

Adaptation of hydrothermal and co-precipitation syntheses particular to intrinsic, bi-metallic inorganic oxide coatings over other known material fabrication types, contrast to costly alternatives, such as vapor deposition methods [95, 96, 97, 98, 99]. Ensuring consistent absorptance quality at elevated temperatures and a facile fabrication method, adapting a cost-effective, flexible experimental synthesis performed to correlate spectrally-selective behavior and spectrally-dependent material texturing is crucial.

Cao et. al. [1] evaluates performance ratings of ceramic-metal composites as depicted in Table.3.1, based on those candidate materials that report mid-range temperature operation, $T = 500^\circ$C.
Matrix | Metal Insert | Preparation | Substrate         | Tested Stability  | $\alpha$ | $\epsilon$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Al$_2$O$_3$</td>
<td>W</td>
<td>Sputtering</td>
<td>Stainless Steel</td>
<td>580° C vacuum</td>
<td>0.939</td>
<td>0.12*</td>
</tr>
<tr>
<td></td>
<td>Pt</td>
<td>Evaporation</td>
<td>Stainless Steel</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SiO$_2$</td>
<td>Ni</td>
<td>Evaporation</td>
<td>Quartz</td>
<td>500° C vacuum</td>
<td>0.9</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>Ni</td>
<td>Sputtering</td>
<td>Aluminum</td>
<td>-</td>
<td>0.96</td>
<td>0.14*</td>
</tr>
</tbody>
</table>

Table 3.1: Mid-range temperature cermet material properties, reproduced from Cao et. al. [1]

Additionally, Bermel et. al. [2] and Chester et. al. [3] report on the quality of molybdenum in the same matrix, preparation, and substrate types, as depicted in Table. 3.2.

Matrix | Metal Insert | Preparation | Substrate | Tested Stability  | $\alpha$ | $\epsilon$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Al2O3</td>
<td>Mo</td>
<td>Sputtering</td>
<td>Stainless Steel</td>
<td>1h@800° C vac</td>
<td>0.92</td>
<td>0.16 (80°C)</td>
</tr>
<tr>
<td></td>
<td>Mo</td>
<td>Sputtering</td>
<td>Stainless Steel</td>
<td>2h@800° C vac</td>
<td>0.92</td>
<td>0.19 (80°C)</td>
</tr>
<tr>
<td>SiO2</td>
<td>Mo</td>
<td></td>
<td>Quartz</td>
<td>580° C vac</td>
<td>0.94</td>
<td>0.13 (580°C)</td>
</tr>
<tr>
<td></td>
<td>Mo</td>
<td></td>
<td>Quartz</td>
<td>2h@800° C vac</td>
<td>0.95</td>
<td>0.075 (80°C)</td>
</tr>
</tbody>
</table>

Table 3.2: High-temperature inorganic material properties in oxide matrix, reproduced from Bermel et. al. [2] and Chester et. al. [3]
Extending this to multi-layer cermet structures, Selvakumar et. al. [4] reviews comparable materials, though they are placed on quartz, molybdenum, and stainless steel substrates by physical vapor deposition methods. Experimental findings from an array of cermet coatings demonstrate increasing sensitivity to oxidation above \( T = 500^\circ C \), causing performance degradation [100].

<table>
<thead>
<tr>
<th>Material</th>
<th>Substrate</th>
<th>Deposition</th>
<th>Tested Stability</th>
<th>( \alpha )</th>
<th>( \epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo-SiO2</td>
<td>Quartz</td>
<td>Sputtering</td>
<td>2hr@800(^\circ)C vac</td>
<td>0.95</td>
<td>0.097 (80(^\circ)C)</td>
</tr>
<tr>
<td>Mo-Al2O3</td>
<td>Molybdenum</td>
<td>Vacuum Evap</td>
<td>500hr@920(^\circ)C vac</td>
<td>0.85</td>
<td>0.11 (500(^\circ)C)</td>
</tr>
<tr>
<td>Mo-Al2O3</td>
<td>Stainless Steel</td>
<td>Sputtering</td>
<td>2hr@500(^\circ)C vac</td>
<td>0.91 - 0.93</td>
<td>0.19-0.27 (80(^\circ)C)</td>
</tr>
<tr>
<td>Ni-Al2O3</td>
<td>Quartz</td>
<td>Evaporation</td>
<td>100hr@500(^\circ)C air</td>
<td>0.96</td>
<td>0.22 (150(^\circ)C)</td>
</tr>
<tr>
<td>W-Al2O3</td>
<td>Stainless Steel</td>
<td>Sputtering</td>
<td>30day@580(^\circ)C vac</td>
<td>0.93</td>
<td>0.10 (400(^\circ)C); 0.14 (550(^\circ)C)</td>
</tr>
</tbody>
</table>

Table 3.3: Listed high-temperature solar selective coatings: single-, double-layered cermets; multilayer and tandem absorbers [4]

From the summary of materials provided in Tables.3.1-3.3, the recommended candidate materials to target for synthesis include:

1. “Black Tungsten” \( W - WO_x \) \( \alpha / \epsilon (800^\circ C) = 0.83/0.15 \) (Temperature cited as vacuum stability)
2. Hybrid cobalt metal-oxides $M_x\text{Co}_{3-x}\text{O}_4$ (M = Ni, Cu, Mn, and Fe) and Hybrid Copper Metal-Oxide $M_x\text{Cu}_{3-x}\text{O}_4$ (M = Ni, Cu, Mn, and Fe)

3. Four (4) possible cermet structures: $W\text{–Al}_2\text{O}_3$, $W\text{–SiO}_2$, $Mo\text{–Al}_2\text{O}_3$, $Mo\text{–SiO}_2$

Based on the success of cuprous materials from previous studies, more in-depth investigation and characterization of bimetallic oxide species, disavowing complex synthesis routes of composite structures, and reports of costly manufacturing for molybdenum and tungsten, returned the following findings for materials that predominantly feature cuprous-based oxide materials:

A work by Kaluza et. al [101] demonstrated the spectral-selectivity of $\text{CuFeMnO}_4$-based coatings synthesized by sol-gel methods, and a more facile approach recently derived by Kim et. al [30] reported the hydrothermal synthesis of tandem-structured Cu–Fe–Mn oxides that maintained 90% absorbance conversion efficiency. Cobalt black, sustainable at $T = 600$°C, was synthesized by Chidamaram et. al. [102] via spray-pyrolysis with an effective integrated absorptance of 93% and hemispherical emissivity of 14%. The synthesis of cobalt oxides [103] to make black cobalt [29] was elaborated upon by Amri et. al. [104], where bi-metallic cobalt oxides (the best absorptance response being $\text{Cu}_x\text{Co}_y\text{O}_z$-based) were synthesized by sol-gel and dipcoating methods to achieve trends to spectral selectivity. A $\text{Cu–Co–Mn–Si}$ mixed spinel type, characterized by Joly et. al. [105], reported a 96% absorptance and 12% emittance.
Chemical Synthesis & Coating Fabrication

As an extension of inorganic oxide nanoparticle syntheses previously characterized for solar absorber coating potential [30, 106], such as CuO, Co₃O₄, CuCr₂O₄, and CuFeMnO₄, experimental syntheses targeted in this study investigated bi-metallic cuprous oxide species: copper cobalt oxide, copper manganese oxide, and copper ferric oxide. Based on the air-stability of the oxide constituents, these materials were reacted to spinel types (isometric minerals of the form A²⁺B³⁺O₄²⁻) via a modified hydrothermal synthesis from comparable material studies [107].

Moon et. al. reports the synthesis of cobalt oxide nanoparticles that have desirable solar absorbing properties, while survivable at high temperatures based on extended thermocycling of \(T = 750\, ^\circ\text{C}\) annealing [31]. Based on preliminary co-precipitation and hydrothermal reaction conditions converging to phase-stable copper cobalt oxide and copper manganese oxide spinels, \(\text{Cu}_x\text{Co}_{3-x}\text{O}_4\) and \(\text{Cu}_x\text{Mn}_{3-x}\text{O}_4\), deriving nanoparticles from an adapted reaction described in Karas et. al. [108], inorganic chlorohydrate precursors were reacted to metal-oxide nanoparticles that were later annealed and ball-milled.

Of the synthesized materials that featured strong solar absorptance potential, novel cuprous-based spinels (\(\text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4\) and \(\text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4\)) were formed from the following reactions by hydrothermal syntheses and co-precipitation methods as found in Table.3.4, as the configurations result from co-precipitation and hydrothermal induced reactions, purification, and annealing treatments.
### Table 3.4: Chemical reaction detail of precursor inorganic chlorohydrates resulting in desired cuprous spinel nanomaterials

<table>
<thead>
<tr>
<th>Reaction 1</th>
<th>Reaction 2</th>
<th>Reaction 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{CuCl}_2 \cdot 2\text{H}_2\text{O} + \text{CoCl}_2 \cdot 6\text{H}_2\text{O} + 4\text{NaOH} \rightarrow \text{Cu(OH)}_2 + \text{Co(OH)}_2 + 8\text{H}_2\text{O} + 4\text{NaCl})</td>
<td>(\text{Cu(OH)}_2 + \text{Co(OH)}<em>2 + \text{O}<em>2 \rightarrow \text{Cu}</em>{2}\text{Co}</em>{3-x}\text{O}_4 + \text{H}_2\text{O})</td>
<td>(\text{CuCl}_2 \cdot 2\text{H}_2\text{O} + \text{MnCl}_2 \cdot 4\text{H}_2\text{O} + 4\text{NaOH} \rightarrow \text{Cu(OH)}_2 + \text{Co(OH)}_2 + 6\text{H}_2\text{O} + 4\text{NaCl})</td>
</tr>
</tbody>
</table>

**Phase I: Co-Precipitation & Hydrothermal Nanoparticle Synthesis**

Varying initial concentrations and amounts of precursor chlorohydrates used produce a variety of uni-metallic and bi-metallic spinel configurations. For instance, reaction of copper chloride dihydrate and cobalt chloride hexahydrate can not only produce \(\text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4\) (a desired result based on stability when exposed to high temperatures), but also \(\text{CoCo}_2\text{O}_4\), \(\text{Co}_3\text{O}_4\), and \(\text{Cu}_{0.27}\text{Co}_{2.73}\text{O}_4\), the latter species generally nettlesome to stabilize when exposed to high working temperatures. Annealing conditions for \(\text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4\) and \(\text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4\) nanoparticles were adapted based on the phase diagrams provided in [109] and [110], respectively.

Using precursor inorganic chlorohydrates, 1 M \(\text{CuCl}_2 \cdot 2\text{H}_2\text{O}\) and 1 M \(\text{CoCl}_2 \cdot 6\text{H}_2\text{O}\) in a 1:60 volume ratio for producing \(\text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4\) spinels or 1 M \(\text{CuCl}_2 \cdot 2\text{H}_2\text{O}\) and 1 M \(\text{MnCl}_2 \cdot 4\text{H}_2\text{O}\) in a 1:1 volume ratio for producing \(\text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4\) spinels, the formed solution can be vigorously stirred for several hours to ensure uniformity. Fig.3.1 demonstrates the preparation, where solid \(\text{CuCl}_2 \cdot 2\text{H}_2\text{O}\) is prepared in a 1M concentration.
Figure 3.1: Precursor inorganic chlorohydrates, such as $1 \text{M} \text{CuCl}_2 \cdot 2\text{H}_2\text{O}$, prepared in 1M concentrations for co-precipitation preparation

The mixed solution was then co-precipitated with the addition of $10 \text{M NaOH}$ to reach a $10 < \text{pH} < 11$ range. Contents from this reaction were then placed in a PTFE-lined autoclave, treated under hydrothermal synthesis at $T = 200 ^\circ \text{C}$ for 20 hours. Succeeding this reaction, additional aqueous contents were decanted and the remaining precipitates were cleaned in a centrifuge using ultra-purified deionized water, electrolytically rated at $18 \text{M} \Omega$. The cleaning cycles were repeated several times for $t = 30 \text{min}$ at 4000 RPM. The materials were then placed in a air freezer
and a vacuum freezer for $t = 12\, \text{h}$, where the expected nanoparticle sizing would be of diameters $D_{\text{np}} \leq 50\, \text{nm}$. Finally, the samples are annealed in a furnace at a temperature of $T = 750\, ^\circ\text{C}$ for $t = 5\, \text{h}$ to induce phase-stability conditions and particle agglomeration at specific mesoscale sizing, approximately $D_{\text{np}} \approx 300\, \text{nm}$. Fig.3.2 depicts the intermediary step of placing clean, dried nanoparticles into a ceramic crucible so as to anneal nanoparticles securely.

Figure 3.2: Inorganic oxide nanoparticles placed in a crucible, preceding $t = 5\, \text{h}$ annealing treatment for inducing phase stability
Phase II: Absorber Coating Port  Nanoparticles of the prepared cuprous-oxide material can be ported to a slurry coating capable of application to metallic substrates by spray-coating, so as to simplify the material deposition process onto a solar receiver.

The nanoparticles are combined with a combination of P/80X SILIKOPHEN resin (SiO$_2$ binder) and solvent (‘XI’, a 3:1 ratio of xylene and isobutanol), according to the configurations in Table.3.5. Specifically, the XI solvent is probe sonicated with the P/80X resin for $t = 30$ min, and probe sonicated again for $t = 30$ min by adding Cu$_{0.15}$Co$_{2.84}$O$_4$ or Cu$_{1.5}$Mn$_{1.5}$O$_4$ nanopowders. Optionally, select sizes of sacrificial polymer beads (SPBs) to induce surface texturing can be added to the solution, in an intermediary probe sonication step also run for $t = 30$ min.

Surface texturing methods applied to the absorber material can significantly boost overall spectral absorptance – the surface morphology of the material would differ prominently depending on the surface texturing process implemented, with correlation to the spectral absorptance characteristic. From the cuprous oxide-based nanoparticles synthesis, the produced materials, when ported to a viscous slurry via a silicon dioxide (SiO$_2$) binder, were optionally treated with sacrificial polymer beads (SPB) made of poly(methyl-methacrylate) of diameter $D_{SPB} = 1–5 \mu\text{m}$ to explore possibilities for effectively surface-texturing different light-trapping geometries.

Suggested recipes for stabilizing these final candidate materials are included in Table.3.5.
### Sample Name

<table>
<thead>
<tr>
<th>Sample Name</th>
<th>XI Solution (mL)</th>
<th>P/80X Resin (g)</th>
<th>Added SPB (g)</th>
<th>Nanopowders (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cu$<em>{0.15}$Co$</em>{2.84}$O$_4$ (Bare)</td>
<td>2.50</td>
<td>0.23</td>
<td>–</td>
<td>1.00</td>
</tr>
<tr>
<td>Cu$<em>{0.15}$Co$</em>{2.84}$O$_4$ (w/SPB)</td>
<td>2.50</td>
<td>0.45</td>
<td>0.175 + 0.025</td>
<td>1.00</td>
</tr>
<tr>
<td>Cu$<em>{1.5}$Mn$</em>{1.5}$O$_4$ (Bare)</td>
<td>3.00</td>
<td>0.25</td>
<td>–</td>
<td>1.00</td>
</tr>
<tr>
<td>Cu$<em>{1.5}$Mn$</em>{1.5}$O$_4$ (w/SPB)</td>
<td>3.00</td>
<td>0.60</td>
<td>0.100 + 0.100</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 3.5: Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ reaction detail, porting nanopowders to a viscous slurry solution capable for spray-coating.

### Phase III: Deposition

In contrast to sol-gel/dip-coating methods for sample preparation [111, 112, 113], a facile spray-coating method onto corrosion-resistant Inconel-625 coupons was employed in the final phase of the synthesis to optimize for low-cost, energy-efficient processing capable of bulk-scale manufacturability. While the proof-of-concept for qualifying materials as feasible for high-temperature system operation or spectrally-selective response are generally produced by sputter-coating or electroplating methods, this work demonstrates spray-coated samples with improved durability, measurement repeatability, and reliability – by the proposed synthesis methods, this cost-effective avenue is tenable for CSP system longevity. In upscaling, process optimizations for the reduction of waste byproducts, low-cost of process reactants, and ease of synthesis while producing high-purity materials were implemented.

Succeeding deposition by spray-coating onto sandblasted, corrosion-resistant Inconel-625 coupons at $P = 40$psi, the samples are cured on a hot plate in the successive three
steps, as depicted by Fig.3.3: 1) $T = 125^\circ C$ at $t = 2\ h$, 2) $T = 250^\circ C$ at $t = 2\ h$, and finally 3) $T = 500^\circ C$ at $t = 1\ h$.

Figure 3.3: Spray-coated Cu$_{0.15}$Co$_{2.84}$O$_4$- and Cu$_{1.5}$Mn$_{1.5}$O$_4$-type samples onto Inconel-625 substrates undergoing curing

Post-curing, the final result at cool-down to ambient conditions is depicted in Fig.3.4, demonstrating sample sizing, where 1”x1” substrates are used for spectral reflectance measurements and 0.5”x0.5” samples are used for x-ray diffraction and microscopy data acquisition.
Material Analyses & Instrumentation

Qualitatively, synthesized materials are analyzed for topological and compositional structuring by optical microscopy, Field-Emission Scanning Electron Microscopy (FE-SEM), and profilometry measurements. X-Ray Diffraction (XRD) and Energy-Dispersive X-Ray Spectroscopy (EDS) can approximate the quantitative extent for the stoichiometric balance of spinel material achieved from the reaction syntheses. The total spectral absorptance of a material is analyzed by optical measurements taken in the ultraviolet, visible, and near-infrared wavelength regions; Fourier Transform Infrared Spectroscopy (FTIR) allows for determining spectral absorptance in the mid-to-far
infrared region.

**Microscopy (Optical, SEM/FESEM)** While the resolution of the human eye approximates roughly 0.2 mm under optimum conditions, optical microscopy and FE-SEM can image 0.5 µm and 1 nm–3 nm resolutions, respectively.

The latter FESEM instrument (depicted in Fig.3.5) is used to reveal improved topographical surface detail, detect sub-surface information, and detect compositional differences. Per this instrument, increasing accelerating voltage of electrons bombarding the sample will tend to reduce background noise up until saturation conditions on the detector, where structural details are reduced on the surface. Additionally, by increasing the electron count on Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ samples that are sputter-coated with a thin gold coating to enhance conductivity through the sample (Fig.3.6), charging artifacts will be built up within the insulation layers below.
Figure 3.5: Field-Emission Scanning Electron Microscopy (FESEM) used in this study (JEOL JSM-6700F)
In adjusting the FESEM working distance to focus, the depth of field is simultaneously adjusted so as to select the image focus. Based on the sample and detector placement relative to the incident electron beam, the spot size, or beam cross-section incident on the sample, introduces possible aberrations that may be present on the image, relating to spatial errors incurred in the absorber morphology (x- and y-errors). A summary of prominent optical aberrations from FESEM include:

- **Spherical Aberration:** when energy diffracts against lens systems edges that are larger distance from the optical axis, convergence varying focal distances occur compared to paraxial energy

- **Chromatic Aberration:** at low voltages or high resolutions, differences in photon
energies (or wavelengths) subtend different refraction angles

- Astigmatism: an off-axis aberration where independent tangential and sagittal ray fans allow for focusing errors in independent orthogonal directions.

A sample of interest is prepared according to the aforementioned guidelines, with sputter-coating enabled to make the surface topology conductive in the SEM. It is placed on a special stage and secured with carbon tape. When inserted into the SEM, the interior chamber has to be evacuated to vacuum, as electrons cannot travel through air. As referenced in the theory above, the accelerating voltage of the electrons is adjustable, typically operating in the regime of $V = 15 \text{kV–}20 \text{kV}$. Electrons pass through the condenser and objective lens components, and scan coils allow emitted electrons to hit the sample specimen surfaces in order to obtain an image from a detector through scattered electrons – backscattered electrons of higher energy and secondary electrons that release from the sample specimen with much lower energy than the backscattered electrons. Backscattered electrons, being very sensitive to surface structure, provide topographic information (COMPO and TOPO Detectors); in contrast, the secondary electrons that penetrate the sample at a much deeper level can provide information of sample composition at lower resolution (SEI Detector). Different values for the voltage, working distance, aperture specification, and spot size are fundamentally tied to the SEM Brightness equation in Eq.3.1:

$$ B = \frac{4i_p}{(\pi D_p \alpha_p)^2} $$ (3.1)
where brightness $B \, [(A/cm^2 \, sr)]$ is linearly proportional to acceleration voltage – it is dependent on the probe current $i_p \, [A]$, the probe diameter $D_p \, [cm]$, and the beam divergence angle $\alpha_p \, [rad]$. Hence, a smaller beam diameter on the specimen can be implemented with larger achievable brightness so that a signal is still present on the detector. Brightness is related to optical throughput (étendue), and is conserved throughout the system by fundamental laws of radiometry described in Chapter 2, and is related to the second law of thermodynamics.

**X-Ray Diffraction (XRD) & Energy Dispersive X-Ray Spectroscopy (EDS)**

This set of material analysis techniques are based on spectroscopy (EDS) and identifying phase content from atomic lattice configurations (XRD); where both analyses are governed by the grating equation: $n\lambda = 2d\sin\theta$. For a periodic arrangement of atoms, the path difference is equal to some integer of light. The periodic displacement of the planes is given by the variable $d$. This is the basis for many diffraction and interference phenomena – birefringence, interferometry, etc.

XRD, in particular, provides material space group information, due to specific atomic arrangements. Based on quantum mechanics effects, partial occupancy of atomistic energies can be found at different sites, and varying stoichiometry can occur by the different sites being occupied in the unit cell. X-ray scattering is dependent on the valence electron density and repelled photon. Most of the scattering is dominated by the cation species – hence, for cuprous oxide spinel species, the oxygens won’t be apparent. The intensity of the peaks is determined by the specific phase
content of the element, defined by crystallographic planes – all crystal structures have the same shape, but not necessarily the same intensities. Intensities tell relative electron density in the unit cell. Which is why Rietfeld refinement is employed. Comparisons to material data in the International Crystal System Database (ICSD) are made to assess the material purity, and support classification of atomistic composition (stoichiometry), crystal structure (applied group theory), electronic properties (band gap information), and also elements of surface morphology (topology). XRD data of nanoparticles and coatings are presented in Fig.3.10-3.11, respectively. As a means of additional verification for uniform stoichiometry present in the sample, Energy-Dispersive X-Ray Spectroscopy measurements are taken on the sample, as presented in Fig.3.13.

**Spectral Absorptance** Spectral absorptance measurements were taken according to the setup depicted in Fig.3.7, consisting of a Newport® TLS-250QU tungsten-halogen light source, collimating optics, filter wheel, CS130-USB-3-FH Cornerstone® 130 scan monochromator, and Newport® 70682NS PTFE-coated integrating sphere. The optical setup is based on comparable versions [114, 115] that test spectral reflectivity, and thereby absorptivity based on null transmittance occurring through the sample. For all measurements, the figure-of-merit (FoM) data as described in Eq.2.32 was determined by specifying calculation boundaries from \( \lambda = 300-1650 \) nm wavelengths. While the tungsten-halogen source, monochromator, and detector are rated at a starting wavelength of \( \lambda = 200 \) nm, larger error ranges between \( \lambda = 200-300 \) nm
due to lower radiant exitance of the tungsten-halogen source, spectral reflectivity measurements tabulated in the UV-VIS spectral range with a Si-photodiode detector (Newport 918D-UV-OD3R) and in the NIR range with a InGaAs-photodiode detector (Newport 918D-IG-OD3R), spanned a range from $\lambda = 300$–1650nm, the calculated boundaries of $[\lambda_1, \lambda_2]$ when solving Eq. 2.32 for the figure-of-merit efficiency.
Figure 3.7: CAD detail for optical absorptance setup, consisting of a tungsten halogen source, scan monochromator, integrating sphere, and photodiode detectors
To obtain spectral reflectance profiles, the spectral profile output from the monochromator, approximated as a narrow Gaussian window for each set wavelength, subtends \( \lambda = 200 \text{ nm–}1650 \text{ nm} \). The spectral reflectance of illuminated samples are determined based on Eq.3.2:

\[
R = \frac{P_{\text{samp}} - P_{\text{cone}} - P_{\text{dark}}}{P_{\text{bright}} - P_{\text{cone}} - P_{\text{dark}}} \tag{3.2}
\]

where \( P_{\text{samp}}, P_{\text{bright}}, \text{and} P_{\text{dark}} \) are the detected spectral power \( P(\lambda) \) due to respective light irradiant on the sample, PTFE bright standard surface, and a dark power measurement when the TLS-250QU tungsten-halogen source is deactivated, accounting for leakage by stray light. Additionally, due to light outside of the target irradiance profile that diffracts around the aperture placed after the scan monochromator, power contributions taken with a baffle cone (approximated to be a pure absorber within the target area) are also nullified from the sample power to bright power standard ratio. When the cone cannot be employed for repeated measurements, consistent measurement with the comparable aperture sizing and sample placement allows for a direct solve for \( P_{\text{cone}} \) detected power to clear from additional detector readings, based on the derivation from Eq.3.3-3.5 and approximated reflectance \( R_{\text{app}} \):

\[
R_{\text{app}} = \frac{P_{\text{samp}} - P_{\text{dark}}}{P_{\text{bright}} - P_{\text{dark}}} \tag{3.3}
\]

\[
\frac{R}{R_{\text{app}}} = \frac{(P_{\text{samp}} - P_{\text{dark}}) - P_{\text{cone}}}{(P_{\text{samp}} - P_{\text{dark}}) - P_{\text{cone}}} \times \frac{(P_{\text{samp}} - P_{\text{dark}})}{(P_{\text{samp}} - P_{\text{dark}})} \tag{3.4}
\]
\[
P_{\text{cone}} = \frac{(R - R_{\text{app}})(P_{\text{samp}} - P_{\text{dark}})(P_{\text{bright}} - P_{\text{dark}})}{R(P_{\text{samp}} - P_{\text{dark}}) - R_{\text{app}}(P_{\text{bright}} - P_{\text{dark}})}
\]  

(3.5)

Results from the optical measurements are depicted in Fig.3.12, depicting the absorptance improvement of novel synthesized solar absorber coating materials when surface-texturing enhancements are implemented.

Results & Discussion

Field-emission scanning electron microscopy (FESEM) data of Cu_{0.15}Co_{2.84}O_{4} and Cu_{1.5}Mn_{1.5}O_{4} nanoparticles and coatings are presented in Fig.3.8-3.9, respectively. Fig.3.8 features Cu_{0.15}Co_{2.84}O_{4} and Cu_{1.5}Mn_{1.5}O_{4} nanoparticles as-prepared succeeding the end of the preliminary co-precipitation and hydrothermal reaction routes in Fig.3.8A and Fig.3.8C respectively, while extended annealing of nanoparticles \((T = 750^\circ C \text{ at } t > 1000 \text{ h})\) are presented for Cu_{0.15}Co_{2.84}O_{4} and Cu_{1.5}Mn_{1.5}O_{4} in Fig.3.8B and Fig.3.8D, respectively. Likewise, Fig.3.9 depicts Cu_{0.15}Co_{2.84}O_{4} coatings (A-D) and Cu_{1.5}Mn_{1.5}O_{4} coatings (E-H). Regular coating preparations (A,B,E,F) differ from surface-textured counterparts (C,D,G,H) with embedded sacrificial polymer beads to induce surface texturing. Images from the first column (A,C,E,G) present as-prepared coatings that differ from coatings exposed to extended annealing (B,D,F,H; \(T = 750^\circ C \text{ at } t > 1000 \text{ h})\).
Figure 3.8: Field-emission scanning electron microscopy (FESEM) images comparing both $\text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4$ and $\text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4$ cuprous-based nanoparticles.
Figure 3.9: FESEM of Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ solar absorber coating materials
As-prepared Cu_{0.15}Co_{2.84}O_4 nanoparticles are smaller (100 nm–200 nm) than the Cu_{1.5}Mn_{1.5}O_4 versions. The latter nanopowders exhibit flake-like structures that extend to 1 µm, under larger particle agglomeration, both XRD and EDS measurements confirm that the material is not subject to significant phase changes. In Fig. 3.10, no visible changes to the peak amplitudes occur when compared to spinel crystal information file (CIF) reference peak counterparts.

Various XRD curves are provided at different annealing exposure and reaction conditions – increased high-frequency content (noise) in an XRD curve correlates to the material crystallinity becoming more amorphous (a decrease in the periodicity of the atomic arrangements decreases deviated radial scatter events). Upon observation, the crystallinity of the material is fairly preserved, as peak types do not experience extraneous sources of noise or feature significant offsets from CIF references.
Figure 3.10: XRD of cuprous-based nanoparticles

Copper Cobalt Oxide Nanoparticles
- Cu_{0.15}Co_{0.84}O_4 CIF Reference Peaks
- Cu_{0.15}Co_{0.84}O_4 As-Prepared Nanoparticles 5hr@750°C
- Cu_{0.15}Co_{0.84}O_4 Extended Annealed Nanoparticles 450hr@750°C
- Cu_{0.15}Co_{0.84}O_4 Extended Annealed Nanoparticles 1050hr@750°C

Copper Manganese Oxide Nanoparticles
- Cu_{1.5}Mn_{1.5}O_4 CIF Reference Peaks
- Cu_{1.5}Mn_{1.5}O_4 As-Prepared Nanoparticles 5hr@750°C
- Cu_{1.5}Mn_{1.5}O_4 Extended Annealed Nanoparticles 450hr@750°C
- Cu_{1.5}Mn_{1.5}O_4 Extended Annealed Nanoparticles 1050hr@750°C

Relative Intensities

Diffraction Angle (2θ Degrees)
Figure 3.11: XRD of cuprous-based solar absorber coatings

Copper Cobalt Oxide Coating Samples

- Cu$_{0.15}$Co$_{2.85}$O$_4$ CIF Reference Peaks
- Bare Sandblasted Inconel-625 Substrate
- Cu$_{0.15}$Co$_{2.85}$O$_4$ As-Prepared Coating 5hr@750°C
- Cu$_{0.15}$Co$_{2.85}$O$_4$ Extended Annealed Coating 450hr@750°C
- Cu$_{0.15}$Co$_{2.85}$O$_4$ Extended Annealed Coating 1050hr@750°C

Copper Manganese Oxide Coating Samples

- Cu$_{1.3}$Mn$_{1.7}$O$_4$ CIF Reference
- Bare Sandblasted Inconel-625 Substrate
- Cu$_{1.3}$Mn$_{1.7}$O$_4$ As-Prepared Coating 5hr@750°C
- Cu$_{1.3}$Mn$_{1.7}$O$_4$ Extended Annealed Coating 450hr@750°C
- Cu$_{1.3}$Mn$_{1.7}$O$_4$ Extended Annealed Coating 1050hr@750°C
For as-prepared samples that have not yet been treated at extended annealing conditions for later reliability testing, both Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ species of coatings with embedded SPBs depict improved spectral absorptance, as compared to standard coatings bereft of surface texturing improvements. Based on the nanopowder synthesis, extended annealing for the first $t = 50$ h improved the coatings’ spectral absorptance, as well as that of the Pyromark reference. As particles agglomerate over time, modifications to the spectral absorptance are anticipated at higher spatial frequencies; surface texturing for light-trapping structures modify transport properties due to larger pore sizes being formed in specific configurations if portions of the material are ruptured. SPBs present at sites, if used in the material, may expand according to portions that are exhumed. Based on three annealing time steps in Fig. 3.12, spectral reflectivity over $\lambda = 200\text{ nm–}300\text{ nm}$ wavelengths increased, correlating to a decrease in sample absorptance.
Figure 3.12: Optical Data
Figure 3.13: Energy Dispersive X-Ray Spectroscopy (EDS) of Coatings Materials (Pyromark 2500 and developed Cu$_{1.5}$Mn$_{1.5}$O$_4$ coating)

Synthesized nanopowders in Fig. 3.8 for Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ feature particle sizes that ranged from $D = 100\text{ nm} - 2\text{ \mu m}$ with agglomeration in post-
annealing treatments measured minimal sizing at $D = 500\,\text{nm}$.

Based on the material coating port in Fig.3.9, stable spinels were maintained without noticeable coating delamination, even succeeding repeated thermal cycling. FESEM morphology for the macroporous structuring based on chemically-bonded SiO$_2$ supported the resistance of coating delamination, as compared to associated raw nanoparticles. Inconel-625 substrates, rated to resist corrosion and oxidation when exposed to elevated pressure and temperature conditions, did not incur any damage at elevated $T = 750\,^\circ\text{C}$ temperatures and ambient pressures. EDS measurements in Fig.3.13 verified minimal corrosion for $t \geq 1000\,\text{h}$ extended annealing on the prepared samples, that also incurred negligible oxidation from both Cu$_{0.15}$Co$_{2.84}$O$_4$ and Cu$_{1.5}$Mn$_{1.5}$O$_4$ species of coating materials on Inconel-625 substrates.

No extraneous contaminants upon various tested sample sites were detected, even elemental constituents from the substrate alloy, as only relative ratios of each bimetallic constituency (Cu:Co and Cu:Mn) were identified for Cu$_{0.15}$Co$_{2.84}$O$_4$ - and Cu$_{1.5}$Mn$_{1.5}$O$_4$-based samples. Cu$_{0.15}$Co$_{2.84}$O$_4$ nanopowders from Fig.3.8, when ported to regular or SPB-embedded coating samples (Fig.3.9), crystalline structures qualitatively showed only minor degradation over extended burn-in testing. The calculated $\eta_{\text{FoM}}$ was stable, on average, for extended annealing conditions beyond $t = 1050\,\text{h}$ of processing. Juxtaposing bare sample coating preparations with SPB-embedded versions, sections of $D = 1\,\mu\text{m}$–$2\,\mu\text{m}$ surface texturing were still apparent for SPB-embedded substrates before and after extended annealing.

Porous structures created as a consequence of SPB types are incurred, based on
varying mean diameters $D_{SPB} = 1.5 \mu m$ and $D_{SPB} = 5.0 \mu m$. Subsequent studies draw attention to the array of light-trapping structures that can maintain high UV-VIS absorptance and reject higher spectral wavelength. Structural geometry in the surface texturing permits incident photons to couple to internally reflected modes based light-matter interactions that feature elastic scattering. The increase of spatial frequencies of particle and material interactions also boosts spectral absorptance capability.

For samples with embedded sacrificial polymer beads for as-prepared structures ($t = 1050$ h extended annealing, featured in Fig.3.9), correlating directly to the optical measurement results in Fig.3.12, where high spatial frequency content of the surface morphology finely-tunes spectral absorptivity.

Morphological comparisons before and after extended annealing demonstrates that surface roughness features for standard Cu$_{0.15}$Co$_{2.84}$O$_4$ samples (Fig.3.9A) is not as effective in absorbing visible light compared to their SPB-embedded counterparts (Fig.3.9C). The coating performance after extended annealing converges to larger apparent surface roughness profiles (on the scale of $300 \text{ nm} - 1 \mu m$) for light absorption in the UV-NIR spectral range (Fig.3.9B,D). Conditions for Cu$_{1.5}$Mn$_{1.5}$O$_4$ coating samples (Fig.3.9E-H), while not immediately apparent in comparing as-prepared samples (Fig.3.9E,G), additionally qualitatively distinguish light-trapping porous structures based on embedded SPBs for samples undergoing extended annealing (Fig.3.9F,H).

In calculating figure-of-merit for extended annealing in Fig.3.14, final SPB-embedded sample versions analyzed succeeding post-annealing treatments, as depicted in Table.3.6, exceeds the performance of prepared Pyromark® 2500 under the same spray-
coating conditions, demonstrating intrinsic absorber coating candidates that feature operating performance increases and maintainability at high-temperatures, as they achieve high UV-VIS absorptivity along with emissivity profiles concomitant with the needs of CSP systems operated at high temperatures and concentration factors. Fig.3.14 shows the calculated figure-of-merit for $t = 150\,\text{h}$ sample measurement iterations when the completed samples underwent extended annealing, as a form of reliability testing.
Figure 3.14: Coating Figure-of-Merit
The results in Table 3.6 reveal the FoM efficiency metrics of fabricated solar absorber coating materials, calculated based on Eq. 2.31 at $T = 750^\circ$C and optical concentration $C = 1000$, over the average of all measurements $\eta_{\text{FoM-AVG}}$ or the final calculated FoM $\eta_{\text{FoM-END}}$. The content of SPBs used (formatted as the mass of $D_{\text{SPB}} = 1.5\,\mu m + 5\,\mu m$, the mass of 1.5µm and 5µm bead sizes respectively) and tested spectral range is also provided.

<table>
<thead>
<tr>
<th>Sample Name</th>
<th>$\eta_{\text{FoM-Avg}}$</th>
<th>$\eta_{\text{FoM-End}}$</th>
<th>Mass of SPBs (g)</th>
<th>Spectral Range (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pyromark 2500</td>
<td>0.8975</td>
<td>0.8920</td>
<td>–</td>
<td>0.3 - 20</td>
</tr>
<tr>
<td>Cu$<em>{0.15}$Co$</em>{2.84}$O$_4$ (Standard)</td>
<td>0.8918</td>
<td>0.8893</td>
<td>–</td>
<td>0.3 - 20</td>
</tr>
<tr>
<td>Cu$<em>{0.15}$Co$</em>{2.84}$O$_4$ (w/SPB)</td>
<td>0.9038</td>
<td>0.9026</td>
<td>0.175 + 0.025</td>
<td>0.3 - 20</td>
</tr>
<tr>
<td>Cu$<em>{1.5}$Mn$</em>{1.5}$O$_4$ (Standard)</td>
<td>0.8890</td>
<td>0.8854</td>
<td>–</td>
<td>0.3 - 20</td>
</tr>
<tr>
<td>Cu$<em>{1.5}$Mn$</em>{1.5}$O$_4$ (w/SPB)</td>
<td>0.9090</td>
<td>0.9048</td>
<td>0.100 + 0.100</td>
<td>0.3 - 20</td>
</tr>
</tbody>
</table>

Table 3.6: Figure-of-Merit (FoM) efficiency calculated for fabricated absorber coatings

The extent of material synthesis and experimental data characterization using most of the aforementioned analysis methods were accepted for publication [108] based on the attached manuscript in App. . Succeeding the experimental development of operationally-competitive solar coating materials, custom simulation routines
tracking solar energy propagation are formulated, to be used in conjunction with experimentally-derived data to validate modeling contributions to this work.
In computational materials science, mesoscale modeling requires coarse graining of atomic distributions to solve a collection of custom interatomic potential compared to ab-initio methods (atomic scale) or continuum methods (macroscale) [116, 117]. Material properties at mesoscale include attention to the following structural characteristics: grain size, texture, precipitate geometries, modulus, toughness, permeability, fatigue behavior, etc., using calculated experimental data such as thermal response, stress/strain history, phase transitions, interface mobilities, and elastic constants [118]. Table 4.1 references the class of known software codes that are ubiquitous for material modeling at different scales, atomic/microscale, mesoscale, and macroscale. They outline, in particular, the following length scales:

- The extent of computational chemistry runtime at atomic scale, using ab-initio methods based on Hartree-Fock formulations or more complex modeling to solve quantum wavefunctions at the atomic level. They can be specified
for energy output based on bond stretch, torsion strain, van der Waals contributions, electrostatics, and ensemble or thermal properties. Inputs material configurations, chemical formulations based on stoichiometric data, and media properties based on material environment can upscale to finite-difference and finite-element analysis equivalent formulations based on the methodology overlap of statistical propagation.

- At larger **macroscopic** levels, finite-element methods [119, 120] are generally the most pervasive for modeling mechanical numerical simulations, computational fluid dynamics, temperature evolution, and harmonic analysis.

- Continuum mechanics [121] are generally used for materials science modeling, applying constitutive relations for computational thermodynamics considerations. Notably, finite-difference time domain (FDTD) methods use numerical approaches to solving Maxwell’s equations by successively alternating incremental changes to electric and magnetic fields due to spatial and temporal propagation:

\[
\frac{\partial \vec{H}(\vec{r}, t)}{\partial t} = -\frac{1}{\mu} \nabla \times \vec{E}(\vec{r}, t)
\]

\[
\rightarrow \frac{\vec{H}(\vec{r}, t + \Delta t/2) - \vec{H}(\vec{r}, t - \Delta t/2)}{\Delta t} = -\frac{1}{\mu} \nabla \times \vec{E}(\vec{r}, t)
\] (4.1a)
\[ \frac{\partial \vec{E}(\vec{r}, t)}{\partial t} = \frac{1}{\epsilon} \nabla \times \vec{H}(\vec{r}, t) \]

\[ \Rightarrow \frac{\vec{E}(\vec{r}, t + \Delta t) - \vec{E}(\vec{r}, t)}{\Delta t} = \frac{1}{\epsilon} \nabla \times \vec{H}(\vec{r}, t + \Delta t/2) \]  

(4.1b)

- In the **mesoscopic** regime, analytic methods to model material transport can be performed with Langevin dynamics [122, 123] and the use of stochastic differential equations [124, 125]. Additionally, Brownian dynamics inspires computational routines in phase-field methods [126] and kinetic Monte Carlo simulations [127], with the ability to account for nonlinear dynamics [128].

**Background & Motivation**

Motivation and relationships among varying computational methods are presented in Table.4.1 – notably, macroscale-based software packages are not adequately adaptive to transport calculations upon microscopy data, as most CAD environments require parametric modeling with NURBS surfaces for maintaining geometric complexity. For this reason, a custom raytracing method is developed in this study based on the presentation of microscopy data leveraging a means of deterministic energy propagation, where the ensemble of atomistic-level interactions are too numerous to differentiate mesoscale topography and thereby calculate energy transport conditions effectively.
Class of Computational Method/Model | Inputs | Outputs | Software Examples
--- | --- | --- | ---
Quantum Chemistry (Electronic Structure Theory, Density Functional Theory, Molecular Dynamics) [ATOMIC SCALE] | Atomic Number, Mass, Valence Electrons, Crystal Structural Configuration w/Lattice Spacing, Atomic Arrangement | Electronic properties, Elastic constants, Free energy vs. structure, activation energies, reaction pathways, defect energies and interactions | VASP, Wien2K, CASTEP, GAMES, Gaussian, a=chem., SIESTA, DACAPO, LAMMPS, DL-POLY

Solid Mechanics, Phase-Field, Constituents [MESOSCALE] | Processing thermal, stress/strain history, phase transitions, interface mobilities, elastic constants | Microstructural characteristics + response: grain size, texture, precipitate geometries, modulus, toughness, permeability, fatigue behavior | OpenPF, MICRESS, DICTRA, Rex3D, OOF, FRANC-3D, DARWIN, Jmat Pro, PrecipiCalc, Amira, Voronoi Cell

Part-level FEA, Finite-Difference, Continuum Models, Neural Nets, Statistical Tests [MACROSCALE] | Part geometry, component loads, environmental considerations, material properties, composition | Temperature distribution, stresses + deformation, electric/magnetic field transitions, optical behavior | ProCast, COMSOL, DEFORM, LS-DYNA, Abaqus, NASTRAN, ANSYS, SYSTAT, MATLAB

Table 4.1: Table outlining the relationships of inputs and outputs properties for different material scaling. (Reproduced in part from Pollock et. al. [5])

This study implements aspects of scalar diffraction theory and Monte Carlo ray-tracing, given their computational robustness, to determine energy transport as it relates to solar coating morphology. Iyengar et. al. [46] cites the usage of Monte Carlo simulations upon laser-textured silicon, noting that the calculation of multiple, successive reflections supports light trapping, and that scattering conditions can be considered as angular deviations from specular reflections events. While the material in this study is assumed to be homogeneous, accounting for the proportions of varying materials would require the dependency of relative spectral absorption at each ray
interaction. Yu et. al. [48] notes the wave considerations of interacting photons upon nanostructured solar cells, based on a previous developed formalism [129] in calculating coupled modes from varying surface structures proportional to material porosity. Essentially, a major hypothesis to the performed studies in this work is that analysis of surface morphology can help to infer additional intrinsic material properties – dependencies from both the ray optics and wave optics interpretation of energy propagation are implemented to leverage structural-dependent spectral absorption. The extent of the simulation refining the experimental synthesis is depicted in Fig.4.1, where a geometrical optics formulation supports the calculation of specular reflectance detail by deterministic Monte Carlo raytracing methods to optimize experimentally-derived solar absorber coatings in a cycle that help experimental refinements.

Figure 4.1: Flow diagram of experimental and simulation processes to optimize spectrally-selective solar absorber coatings based on energy transport phenomenology from continuously-refined surface texturing
The goal of this study will be to further increase the material efficiency metrics of the devised cuprous-based inorganic oxide materials that are used as solar absorber coatings by classifying morphological detail. At varying length scales, optical microscopy supports the topological acquisition of smaller magnifications (10x-500x) to detail low spatial frequency patterning; conversely, field-emission scanning electron microscopy (FESEM) allows for resolving higher spatial frequencies at larger magnifications (1000x-100,000x).

In Fig.4.2, surface profiles of optical microscopy are depicted based on the resulting RGB image produced by a Olympus BX-51 ‘Pseudo-Confocal’ Microscope (4.2A,C respectively). Comparable details are also noted for FESEM imaging (4.2B,[D,E,F]), where the latter image set span composition, secondary-electron backscatter, and topographic detection schemes. The surface profile of both surface plots’ spatial axes (4.2A,B) are set to help visualize the length scaling of notable topological features.
Figure 4.2: Image data, optical and field-emission scanning electron microscopy
Monte Carlo Raytracing

The totality of the Monte Carlo raytracing routines are provided in the appended listings, Listing.6.1-6.7: interpreted code authored in MATLAB, and ported compiled code in C++, developed in part to help improve computational runtime.

An overview of the proposed runtime in summarized in Alg.1, where input microscopy data (fixed at 968x1280 pixels – 1024x1280 pixels total, 56x1280 pixels are at tail of image and constitute metadata) incorporates a ray bundle of 1,239,040 incident rays. A brute-force approach of calculation that permits pseudo-random sampling by Monte Carlo methods implemented in a subsequent step, is to maximize the ray count to the resolution of the microscopy data at all sample points, to allow flexibility in selecting random ray distributions. Notably, this is a computationally expensive calculation performed primarily to evaluate statistical variability among random collections of the completed specular ray calculations. For a ray depth of ten iterations, supplying three rays for each iteration (propagation, normal, specular reflection), a sum total of 37,171,200 rays are calculated for a single SEM image!
Algorithm 1 Brute-force raytracing runtime, compacting listings as presented in App.1

Data: Input SEM image
Result: Returns a struct with ray calculation summaries

Initialization for all pixels do
  propagate from current position
  if ray intersection found then
    determine new calculation parameters based on new ray depth, set new position and new propagation direction
  else
    classify as diverged ray at ray depth, calculation parameters become static
  end
end

Fig.4.4 features visualization of the incident ray bundle on the sample surface, using the function defined in Eq.4.2 to create a surface with both peak variation and flat surface profiles.

\[ f(x, y) = 3(1 - x)^2 \exp\left( -x^2 - (y + 1)^2 \right) - ... \]
\[ 10(x/5 - x^3 - y^5) \exp\left( -x^2 - y^2 \right) - ... \]  
\[ 1/3 \exp\left( - (x + 1)^2 - y^2 \right) \]  

(4.2)

The incident rayset is shown in the direction of [0, 0, -1] vector propagation; rays that interact with the surface colocate ray normals on the surface to calculate the first interaction of reflected rays. In the adjoining histogram, large counts for the elevation angle demonstrate the larger portions of flat area, based on reflected ray
angles close to $\pi/2$ radians elevation angle. Gaps in the elevation angle become an artifact of the sampling conditions of the surface profile as calculated in Eq.4.2, where $x = y = [-8 : 8]$ and a 17x17 pixel count subtends the $x$ and $y$ axes. The input images, of 968x1280 pixel sizing, have ray normals and and the first specular reflection from these points calculated based upon scanning locations of horizontal and vertical nearest neighbors. For reflected rays that do not immediately diverge after the first surface interaction, a specified number of maximum ray interactions govern the continuity of the raytracing calculation. For each pixel, the input image operates on a tessellated version of the input data set (Fig.4.3) to nullify data that would otherwise possibly diverge away from the surface dataset too quickly by surpassing a boundary.

Figure 4.3: Tessellated SEM image for Monte Carlo data calculations, expanding the usable data runtime according to rays that diverge immediately past the boundary confined by the central 968x1280 dataset
A cross section of the microscopy data can be depicted via an image profiling along two points defined in the dataset – the point at initial ray interaction and the range the reflected vector propagates. In determining if the specularly reflected ray interacts with this cross section, a subsequent ray interaction is collected according to interpolating the image intensity from nearest neighbors defined at integer pixel spacing. The routine restarts based on this interaction, else the ray diverges and does not contribute to additional absorption events.

Figure 4.4: Monte Carlo Raytracing Visualization
Supercomputing Hardware Specifications

On a regular single-user workstation computer, this type of brute-force calculation approximates $t \approx 30$ /hour of computation time. A batch job port to a supercomputer cluster described below was able to reduce the brute-force raytracing calculation to $t \approx 5$ /hour. According to the most recent specifications listed on the National Supercomputing Institute (NSCEE) website hosting the hardware:

“The name Cherry Creek was used by Intel when they built and demonstrated the machine at trade shows. We started using the name internally and it just stuck. The original Cherry Creek was used as a demo machine at several trade shows and conferences. It was ranked on the top 100 Green machines and in the top 500 supercomputer lists. The original Cherry Creek (1.0) had 48 nodes. Each node had 2 Intel Xeon E5 - 2697v2 (12 cores each), 128Gb Ram, and 3 Intel Xeon Phi 7120P coprocessors (with 61 cores each). In addition to the 48 nodes above, Cherry Creek 2.0 has an additional: 48 Penguin Computing Relion nodes each with 2 Xeon E5 - 2640v3 (8 cores each), 128Gb Ram, and 4 Intel Xeon Phi 31S1P (with 57 cores each). It also contains 24 Intel manufactured nodes with 2 Xeon E5 - 2697v2 (12 cores each), 192Gb Ram, and 2 Intel Xeon Phi 7120P coprocessors (with 61 cores each). Cherry Creek 2.0 currently has the following capabilities: theoretical peak speed of 495 TFlops/s (Trillion Floating-Point operations per second), total memory: 32.470 TB (TeraBytes), and total scratch storage of 46.32 TB. It is ranked 394 on the June 2015 Top 500 list (http://www.top500.org) and 186 on the June 2015 Green 500 list (http://www.green500.org).”

Raytrace Summary

A summary of the raytracing results is depicted in the array of figures ranging from Fig.4.5-4.15: Boolean values for continuation of the raytracing calculations, power values, vector components, offsets from initial interactions, azimuths and elevations, and the extent of interactions per pixel start position designated as the ‘ray depth’.

The extent of boolean values are depicted in Fig.4.5 to show the count of active rays. At the first interaction (first ray depth), all rays are considered active as they propagate from a ray bundle at a certain incident angle to the surface. Rays that
specularly reflect away from the surface after this first interaction are captured by the second ray depth plot, where the extent of porosity on the sample surface correlates to the rays that remain active based on successive specular interactions; rays that diverge at every iteration are nullified according to their Boolean value. Notably, the ray index is specified by the ray’s pixel value at first interaction. Additional summaries tracking vector offsets in cartesian or spherical coordinates track the extent of ray propagation. Comparably, the ray powers in Fig.4.6 is tabulated as a multiplicative factor of the spectral absorptance capability, where $\alpha = 0–1$ and at successive ray depths, the maximum multiplicative factor based on successive ray interactions is equal to the power factor – this employs the same indexing as the Boolean data. Fig.4.7-4.9 indicates the reflected ray’s vector components based on the $[X,Y,Z]$ cartesian coordinates. Fig.4.10-4.12 show the offsets in pixel dimensions for successive ray interactions according to pixel indices. Notably, at the first ray depth, the smooth gradient according to $[X,Y,Z]$ cross sections represents the original data set, with successive ray depths capturing offsets for rays that feature another interaction – rays that diverge are not calculated, and remain with null values at subsequent raytracing steps. Fig.4.13-4.14 track the specularly-reflected rays’ propagation direction in spherical coordinates, as is useful for generating histograms according to angular deviations, as well as solving for successive ray interactions with image profiling cross sections of data. Lastly, Fig.4.15 gives a summary of the original surface normals calculated from the original microscopy dataset for later characterization, as well as indicating the maximum achieved ray depth, as is useful for previewing the extent of
porosity from the dataset.
Figure 4.5: Monte Carlo Raytracing Summary: Ray Booleans
Figure 4.6: Monte Carlo Raytracing Summary: Ray Powers
Figure 4.7: Monte Carlo Raytracing Summary: Ray x-Direcions
Figure 4.8: Monte Carlo Raytracing Summary: Ray y-Direcitions
Figure 4.9: Monte Carlo Raytracing Summary: Ray z-Direcions
Figure 4.10: Monte Carlo Raytracing Summary: Ray x-Offset
Figure 4.11: Monte Carlo Raytracing Summary: Ray y-Offset
Figure 4.12: Monte Carlo Raytracing Summary: Ray z-Offset
Figure 4.13: Monte Carlo Raytracing Summary: Ray Azimuths
Figure 4.14: Monte Carlo Raytracing Summary: Ray Elevations
Figure 4.15: Monte Carlo Raytracing Summary: Additional Raytrace Data
The extent of the raytracing summary for FESEM data is provided in Fig.4.16, for four distinct images: standard and surface textured forms of both Cu$_{0.15}$Co$_{2.84}$O$_4$- and Cu$_{1.5}$Mn$_{1.5}$O$_4$-based solar absorber coatings. According to the data runtime, in calculating at least ten ray depths, the specular reflection detail from the first ray interaction oftentimes accounts for the ability to classify porosity – for all four samples in Fig.4.16, the active ray count from ray depths 1-2 is largely equal to all subsequent ray depths up to ten interactions. As the percentage of active rays trends to convergence at a ray depth of 10, this runtime of calculation approximates ten ray depths at capturing an accurate amount of energy flux at successive ray interactions. For each sample (standard Cu$_{0.15}$Co$_{2.84}$O$_4$ coating in Fig.4.16A, surface-textured Cu$_{0.15}$Co$_{2.84}$O$_4$ coating in Fig.4.16B, standard Cu$_{1.5}$Mn$_{1.5}$O$_4$ coating in Fig.4.16C, and surface-textured Cu$_{1.5}$Mn$_{1.5}$O$_4$ coating in Fig.4.16D), three different spectral absorption multiplicative factors are tabulated: $\alpha = [0.1, 0.5, 0.9]$. The progression of reflected overall power is given by the closing ray depth, where any absorption value from a wide range is depicted to converge to some known percentage. While each solar absorber coating material, whether Cu$_{0.15}$Co$_{2.84}$O$_4$- or Cu$_{1.5}$Mn$_{1.5}$O$_4$- based, is interpreted to be homogenous and uniform with a specific absorption value, spectral-dependence of the raysets requires a range of varying absorption values, where $\alpha = \alpha(\lambda)$. 
A method to interpret the extent of surface porosity, given preliminary raytracing data, is feasible given the successive offsets from the next ray depth, yet this manner of calculation suffers from an inadequate number of ray paths to accurately determine a transfer function for which wavelengths interact at specific regions, as larger error tolerances exist in correlating subsequent ray offsets with an accurate classification of porosity in certain surface textures. The remaining study investigated image process-
ing techniques to more accurately define porous structures based on preliminary ray interactions, combining this detail to account for the ‘wave’ dependence of photon propagation from incident solar energy.

To summarize, Monte Carlo calculations, especially based on the brute force approach, are computationally expensive to calculate effective detail, especially for high scattering events, or given the possibility of transmissive materials. Initial sampling of the dataset, so as to effectively model porous structures, are of interest to interpret based on the level of both optical and more intricate FESEM imaging.

The resulting study based on the algorithms and methods presented for preliminary simulation applies image processing concepts to tie spectral dependence to the raytrace summary. Statistical manipulations of the surface morphology in accounting for errors as tolerated from Monte Carlo data introduce multiresolution analysis to determine error summaries.
“People who want to improve should take their defeats as lessons, and endeavor to learn what to avoid in the future. You must also have the courage of your convictions. If you think your move is good, make it.”
- José Raúl Capablanca

The previous study demonstrated the use of Monte Carlo raytracing techniques to determine the extent of specular reflections present based on morphological data from microscopy images. The proposed listings, as generated in MATLAB and C++, detail the interpretation of porosity based on the image sampling limited by maximum detector resolution. Yet, interpreting spectral features is an important characteristic that is studied based on proportionality to pore sizing. This final study details the use of image watershedding and multiresolution analysis with wavelet transformations to provide more detailed ray statistics as a means of correlating raytrace calculations to more accurately describe spectral absorptance, for thorough comparisons with experimental data.

Morphological Structures, Texturing, & Porosity

Light trapping occurs due to scatter deviations from multiple reflections - diffuse reflections are described by an elastic scatter function that is more sensitive at shorter
wavelengths, usually interpreted with proportional dependence to the Rayleigh Scattering criteria in Eq.2.38.

Surface Texturing  Methods for accurately inducing surface texturing on a material, such as laser ablation processes [130] or plasmonic nanostructuring [131], while amenable for fine-tuning nanosecond and femtosecond pulse widths [132], generates large material losses due to its role as a post-treatment modification, essentially becoming cost-prohibitive when applying absorptance increases to materials’ surface geometries at bulk scale. Light trapping occurs due to scatter deviations from multiple reflections - diffuse reflections are described by a scatter function that is more sensitive at shorter wavelengths.

Drolon et. al. [65] introduces a particle roughness descriptor as motivated through classifying geological specimens, with surface characterization methods [72] that are compared to Fourier descriptors and fractal dimensioning. Essentially, the calculation of grain roughness, defined as contour irregularities at different scales, is performed at successive resolution levels to interpret scatter detail. Methods to characterize surface contours via image processing transformations support obtaining physical properties: elongation, angularity, roughness, porosity [71]. In comparison, fractal analysis, as a classifier for complex particles, becomes difficult to apply to regular forms (wear) and different types of particles that have the same fractal dimension - cannot successfully differentiate between particle types. Algorithms for interpreting differential geometry are sensitive to noise present on the contour, and thus, the application-dependent
parameters do not permit the ubiquity of the simulation characteristic to be performed on varying structures. Multiresolution analysis, using short-time Fourier transforms or wavelet transforms are not as widespread in pattern recognition since most wavelet coefficients are not shift-invariant, though it can be used based on its data compression applications as a noise removal/filtering technique. The particle roughness classifier must determine a fixed starting point and contour normalizations prior to computing the wavelet transforms.

Buraga-Lefebvre et. al. [68] supports ideas of particle location and classification, as applicable to in-line (Gabor) holography. Based on the implementation of a far-field (Fraunhofer) diffraction pattern, one can determine the 3D location of particles accurately, with applicability of experimental application of spray-coating. Yet, with most algorithms, high particle density regions can be computationally intensive for calculation. Therefore, direct analysis of diffraction patterns without manual focusing can be performed by the use of space-frequency operators. Comparing a classic approach of 3D particle location analysis using the Wigner distribution function, Buraga-Lefebvre et. al. reconstructs contiguous planes of sample volume via an analogous wavelet transform. Comparisons with use of diffraction pattern $t_{zo}(x, y)$ and daughter wavelet $\psi_a(x, y)$, image thresholding will allow for reconstruction of image planes for bulk volume:

$$\psi_a(x, y) = a^{-2} \sin \left( \frac{x^2 + y^2}{a^2} \right)$$  \hspace{1cm} (5.1)
Jiang et. al [133] introduces methods for particle de-noising and error reduction of imaging topology: To properly filter images of microscopy artifacts, embedded change-of-basis on a third-generation wavelet model extracts morphological features using the bi-orthogonal dual tree complex wavelet transform. Attempts to give affine invariance, which is not dependent on reference frame, limit redundancy for efficient computation, and this helps to prevent shift-aliasing for accurate image reconstruction. The custom wavelet model is described in Eq.5.2, with constants $a$ and $d$ defined for low-order basis functions:

$$f(s) = \sum_{k \in Z^2} a_{j,k} \phi_{j,k}(s) + \sum_{b \in B} \sum_{j \leq J} \sum_{k \in Z^2} d_{j,k}^b \psi_{j,k}^b(s)$$ \hspace{1cm} (5.2)

Comparable concepts for inferring particle size distribution by Hryciw et. al. [66] can also be performed by regular Haar Wavelet Decomposition in calculating energy densities for particle types based on their spatial frequencies at different resolutions, finding correlation. Kim et. al. [134] reviews multiresolution analysis for applying image segmentation, discontinuities, feature detection [70]. Of many different types of wavelets used for many applications (Daubechies, Morlet, Haar, Littlewood-Paley, Sombrero, etc.), different types of denoising, discontinuity detection, feature extraction, frequency identification, and data compression implementations are available. Daubechies wavelets, for instance, introduce useful for non-linear structural dynamics, and applicable as a generalization of basic Haar and Morlet wavelets.

For microscopy data susceptible to diffraction offsets, algorithms that calculates
the wavelet transform enact as serial convolutions to a superposition of simpler wavelets. In addition to calculating spatial or temporal frequencies as with the Fourier or Laplace transform, the wavelet transform also dimensionalizes a scale parameter $a$ related to the axial distance $z$ of photon propagation. Original field reconstructed by finding the scale parameter’s optimum value that produces a maximum of the wavelet transform modulus. This can be applied to filtering techniques to extract high-frequency and semi-scaled errors based on microscopy data (Fig.5.1).

Figure 5.1: 2D cross section of material: signal smoothing by noise removal using wavelet transform methods.
Spatial Domain Scanning

Succeeding the experimental synthesis of regular hydrothermal, co-precipitation synthesis of cuprous-oxide solar absorber materials in the previous study, spectral absorptance contributions were measuring in the ultraviolet (UV) to near-infrared (NIR) regions. Additionally, x-ray diffraction (XRD), energy-dispersive x-ray spectroscopy (EDS), and field-emission scanning electron microscopy (FESEM) data were taken for assessment of phase content, stoichiometric interpretation, and material quality. Based on such microscopy data, quantitative methods can be leveraged against other forms of data taken on respective samples, as a means to interpret reaction conditions and create meaningful predictions about optimization trends.

Figs.5.2-5.4 represent a means to classify porosity in the microscopy data; at specific length scales, this can be correlated to the embedded polymer beads used in preliminary reactions when porting cuprous oxide nanoparticles to solar absorber coating materials.
Figure 5.2: Histogram equalization routine, demonstrating the changes incurred on the SEM image for improving feature extraction.

With microscopy data, sample morphology data is quantified based on digital image processing - particle density, distribution, locations, discontinuities, etc. Spatial domain filtering methods and multiresolution analysis using wavelet transforms are applied to infer additional morphological data from microscopy images: i.e. roughness, extent of noise, particle sizing and distribution, and surface contours.
Figure 5.3: Equivalent surface plots, with RGB values in $z$. 
Figure 5.4: Equivalent surface plots, scaled in $z$ for surface height.

Using a Keyence® VK-X250 3D Laser-Scanning Confocal Microscope, profilometry data (Fig.5.5, relative surface errors of $\epsilon \approx 0.5 \text{ nm}$) is leveraged against the surface plots from Fig.5.4, indicating that error tolerances from non-filtered data range from $\epsilon \approx 50 \text{ nm}–80 \text{ nm}$, which can be reduced to approximately $\epsilon \approx 20 \text{ nm}–50 \text{ nm}$ according to filtering effects.
The following data from Fig.5.6-5.7 demonstrates the calculation summary of spatial-domain scanning, interpreting successive thresholded layers of microscopy data for pore sizes by scanning successive x- and y-cross sections of the material surface. Large data counts are treated to equivalently produce pore sizing of the material.
Figure 5.6: Porosity histograms of standard Cu$_{0.15}$Co$_{2.84}$O$_4$- and Cu$_{1.5}$Mn$_{1.5}$O$_4$-based solar absorber coatings, as calculated through spatial-domain scanning of x- and y-cross sections, and averaged to a Gamma distribution.
Figure 5.7: Porosity histograms of SPB-embedded Cu$_{0.15}$Co$_{2.84}$O$_{4}$- and Cu$_{1.5}$Mn$_{1.5}$O$_{4}$-based solar absorber coatings, as calculated through spatial-domain scanning of $x$- and $y$-cross sections, and averaged to a Gamma distribution.
According to Fig.5.6, the range of pore sizes for \( \text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4 \) and \( \text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4 \) differ according to surface-texuring counterparts from Fig.5.7, where a larger number of porous diameters are apparent in the calculation summary. In Fig.5.6, the pore size distribution, as fit with a gamma function, maximizes roughly 13,000 counts of \( D_{\text{pore}} = 0.25 \mu m \) in the standard coating, while Fig.5.7 features less sub-micron pore size counts.

Based on comparisons from raytracing results from Study.2, interpreting the extent of porosity is a useful characteristic for interpreting spectral material interactions – subsequent studies work to demonstrate additional algorithms that were tested to more rapidly determine extent of porosity, especially for use in the real-time Monte Carlo raytracing scenarios.

**Image Processing**

For tracking particle sizes and locations, implementation of the Atherton-Kerbyson modification of the circular Hough Transform is known to track particle sizes at small radii. The routine outputs coordinates (particle locations) and sizes (particle diameters). In Fig.5.8, an implementation of this transform is shown to detect particle sizes accurately at 3px resolutions, yet feature detection with this method must carry significant contrast with a background signal – additionally, while the transform can detect elliptical features, the extent of these transforms on \( \text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4 \) - and \( \text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4 \)-based nanoparticles and solar absorber coatings is less accurate due to the array of porous shapes that are recognizable.
Figure 5.8: For nanoparticles present on a substrate, the Atherton-Kerbyson modification of the circular Hough Transform can track particle locations and sizes.

Formalizing Porous Structuring

More rigorous interpretations to the mesoscopic structuring of solar absorber coatings require image thresholding so as to interpret structuring based on two-dimensional cross sections of the material surface. This is primarily due to the inability of image processing routines to be applied to porous volumes directly, as it is more computationally-efficient to perform segmentation on two-dimensional surfaces, and construct such data to give a description of volume based on contiguous surfaces. The thresholding runtime is depicted in Fig.5.9, where histograms to the standard and surface-textured Cu$_{0.15}$Co$_{2.84}$O$_4$ coating sample FESEM data demonstrates changes to the respective image histograms based on a 50% thresholding condition – pixel RGB
values, as proportional to the sample surface height, are binarized into a boolean value based on sample present at a specific depth. This equates to the ‘z’ cross section of the sample volume.

Figure 5.9: Standard and surface-textured Cu$_{0.15}$Co$_{2.84}$O$_4$ comparing histograms before and after applying image thresholding
A comprehensive algorithm, as adapted and modified from Rabbani et. al. [135], supports real-time identification of various structures based on image watershed segmentation (whose name derives from topographic interpretations of geological drainage divisional structuring). The interpretation of porosity at such cross-sections are sorted according to a label matrix, which helps to identify individual pores and approximate their porous diameter to an equivalent circle – when reconstructed to the material surface, this is a useful way to define porous volumes, supported as a specifier for the Monte Carlo raytracing studies. Leveraging distancing of prominent pore locations and offsets to adjacent pores, minor connections of adjacent pores are nullified to represent porous structures independently. The ‘watershed’ approach is defined according to the comparable watershed geological formations, where for topographic descriptors, porous structures are coordinated with a label matrix to index separated features, based on isolating local minima from surface contours.

An example of its implementation is provided in Fig.5.10 for a Cu$_{0.15}$Co$_{2.84}$O$_4$ coating sample.
Figure 5.10: Watershedding Algorithm on a Cu$_{0.15}$Co$_{2.84}$O$_4$ sample
Using further multiresolution analysis beyond filtering effects, varying structures according to a data reference can be calculated according to the configuration of spatial frequencies present for pore sizing at varying thresholding levels. Identification of pore sizes that can be interpreted according to spherical, trigonal, and varying orders of power functions, as depicted in Fig. 5.11, are reserved for further study based on the interpretation of matching varying pore sizing (Fig. 5.12) with optimal light-trapping geometries based on the extent and angle of incident solar irradiation.

Figure 5.11: Array of Spherical, Trigonal, and Power Functions used for a data reference to compare frequency content of microscopy data
In calculating morphological statistics, accurate topological detail can be used to theoretically infer optical scatter functions – the direct measurement for material scatter can be performed by a bi-directional scattering distribution function (BSDF) measurement to ensure that the simulation detail is working accurately, though the experimental verification of BSDF, especially for novel materials, can be ambiguous without varying surface texturing formations to feature an array of reference comparisons. This study interprets offsets of scatter contributions according to the formalism for Total Integrated Scatter (TIS) [136, 61], where elastic scatter contributions are anticipated to diverge according to Eq.2.38. The current raytracing results depicts averaging for specular reflections, with offsets accounting for ray errors in azimuth and elevation due to scatter conditions.
In calculating successive porous features, histograms upon the four images as calculated in Fig.4.4 for the Monte Carlo raytracing routine describe the detail of porosity for the total material surface, helping to define the volume considerations of the material penetrated by incident solar irradiation (Fig.5.13.)

Figure 5.13: Pore Size Histogram

The comparisons of experimentally-derived spectral absorptance increases are depicted in Fig.5.14 against those derived from simulation in Fig.5.15. Essentially, comparable trends are apparent at higher wavelengths as compared to shorter wavelengths of \( \approx 0.5\% \) error deviations, though the runtime predicts general accuracy of
≈ 1% error deviations.

Figure 5.14: Experimentally-derived spectral absorptance response of \( \text{Cu}_{0.15}\text{Co}_{2.84}\text{O}_4 \) and \( \text{Cu}_{1.5}\text{Mn}_{1.5}\text{O}_4 \)
With the extent of experimental analyses, custom simulations devised according to correlations among data are of value to infer reaction conditions of the initial chemical synthesis. For simple singular compounds with relatively few atomic interactions, quantum chemistry routines can typically solve electron and nuclei potential interactions, with complexity upscaling depending on the amount of subatomic particles considered. In contrast at macroscopic levels, continuum mechanics, considering a
material as a continuous mass rather than a collection of discrete particles, can be a valid approximation for larger length scales. Mesoscale modeling features a ‘worst of both worlds’ consideration sharing in combined properties of atomic scale and macroscopic simulation alike, where material structure generally must be regarded dependent on discrete potentials, but the material structures are too large to consider the plurality of all contributions from atomic-level interactions.
CHAPTER 6

CONCLUSION

“I dreamt all this; never could my poor head have invented such a thing purposely.”
- Richard Wagner

This work has addressed the following research questions, based on successive experimental and simulation studies, by:

1. Motivating an experimental synthesis for development of a novel spectrally-selective solar absorber coating material

2. Developing simulation runtime for energy transport based on solar absorber coating morphology

3. Correlating image processing routines to enable spectrally-dependent energy transport modifications

This research has yielded a statistically-detailed computational formulation of optimizing for energy-efficient material phenomenology of various absorber coaters; much of the design will be supported by a collaborative effort in materials research, data science, and chemistry. The generated optimization routines are to be made freely-available under NSF directives in an open source format, as well as reported
laboratory data for certain example fabrication scenarios and test configurations of interest. These studies, as summarized below, were performed to support improvements to power generation technologies given the application to solar energy, as well as energy-efficient capabilities in other renewable energy sectors.

Study 1: Experimental Synthesis

Preliminary reactions were performed to synthesize and characterize a class of inorganic oxide spinels of the formulaic configuration $A^{2+}B_{2}^{3+}O_{4}^{2-}$ as intrinsic solar absorbers – specifically, copper-cobalt oxide ($Cu_{x}Co_{3-x}O_{4}$), copper-manganese oxide ($Cu_{x}Mn_{3-x}O_{4}$), and copper-ferric oxide ($Cu_{x}Fe_{3-x}O_{4}$) based on literature reviews recommending such spinels from their respective material properties.

This work devised a complete synthesis of inorganic oxide nanomaterials ($Cu_{0.15}Co_{2.84}O_{4}$ and $Cu_{1.5}Mn_{1.5}O_{4}$) as a candidate for spectrally-selective solar absorber coatings.

In qualitatively correlated structuring, topology and porosity to optical scattering events, an estimation for overall spectral absorptivity was confirmed with spectral absorptance measurements performed at room temperature, acknowledging the phase stability of the material structure when undergoing high-temperature thermocycling. While the standard solar absorber coatings developed from $Cu_{0.15}Co_{2.84}O_{4}$ and $Cu_{1.5}Mn_{1.5}O_{4}$ nanoparticles were comparable to commercially-adopted Pyromark® 2500 solar absorber paint, the intermediary step of implementing porous surface geometries (light-trapping structures) allowed for improvements of total spectral absorptive behavior.
Study 2: Simulation & Analysis

Based on the qualitative correlations from Study.1, Monte Carlo raytracing methods, as generally applied to stray light analysis and optical system design, are formulated to assess energy transport in inorganic oxide mesoscopic structuring. When applied to scalar diffraction theory and the calculation of optical surface and volume scatter, this type of analysis is versatile morphological tolerance analysis, error propagation, and filtering [137, 138].

Study 3: Statistical Optimization

Computational approaches to quantifying morphological statistics from microscopy data can work in conjunction with other material analyses to approximate scattering and other macroscopic material properties, supporting synthesis refinement and microstructure optimization for improving the performance of absorber coating technologies.

A large extent of the simulation calculations directly from specular reflections confirmed the greater sensitivity to scattering events at shorter wavelengths, and why simulated spectral absorptance features trends, where longer wavelengths are more accurate to experimentally-derived measurements.

Future Work

Future recommendations for the outcome of studies presented in this work include:

• Development of a computational neural net framework to catalogue varying
topological detail can support experimental inferences in optimizing very specific reaction conditions

- Improvements of optical runtime, as upgrading processing to multicore CPUs supported proof-of-concept of the authored Monte Carlo raytracing algorithms. Since the many rays traced on a single image are not dependent based on every component in a ray bundle, parallelization of raytracing algorithms will vastly improve runtime, helping to make protracted analyses amenable to single advanced workstation computers.

- Adaptation of differential raytracing in comparison to Monte Carlo methods allows for importance sampling essentially determine sequential ray paths for direct radiometric calculations; this class of algorithm hasn’t seen wide-scale adoption and implementation beyond custom optical design codes [139].

- Develop comparisons against more accurate profilometry measurements, such as 3D laser scanning confocal microscope to more accurately determine runtime errors – this, of course, is levied based on high costs of instrumentation. The developed simulation codes are motivated to supplant the need for extended experimental data acquisition.
“Ό, τί δεν συνέβη ποτέ, είναι ό, τί δεν ποιήσαμε αρκετά.”
- Νίκος Καζαντζάκης

“Wat ik te zeggen had, was gezegd.”
- Simeon ten Holt
%% Preliminaries
% Set Directories
cd '/home/dekara/rt'
workDir = '/home/dekara/rt';
% Set Image
img = imread(strcat(workDir,'/semImage.bmp')); img = img(1:964,:);
r0Vec = [0,0,-1];
wavs = 200:5:1650;
absCoeff = 0.5.*ones(size(wavs));
rows = numel(img(:,1)); cols = numel(img(1,:));
solveDepth = 10;
IF = rows-2; jF = cols-2; % Default Pixel Counts
wavLen = 1; % Adaptable to larger wavelengths
tic % Start total computation runtime
% Compute Primary Data
[X,Y] = meshgrid(1:cols,1:rows); % Pixel Normals
[ reflAzim, reflElev, ~, reflX, reflY, reflZ ] = ...
    reflRayFun( vecx, vecy, vecz, r0Vec ); % First Ray Refl
fprintf(['Data preliminaries calculated: Initial normals and'
    ' reflected ray events...']); % reflected ray events...
rays = struct(f1,v1,f2,v2,f3,v3,f4,v4,f5,v5,f6,v6,v7,v8,v9,...
    f10,v10,f11,v11,f12,v12,f13,v13,f14,v14);
clearvars f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12 f13 f14
clearvars v1 v2 v3 v4 v5 v6 v7 v8 v9 v10 v11 v12 v13 v14
clearvars X Y
% Profiling
% rays.power = absCoeff(1).*rays.power; % Fix absorption coefficient later!
if mod(solveDepth,1) ~= 0
    error('solveDepth is not an integer!');
elseif solveDepth > 1
    for iteration = 2:solveDepth
        if ~isa(img,'double') % Convert to Double if Necessary
            img = double(img);
        end
        imgUD = flipud(img); imgUDLR = fliplr(imgUD);
        imgTess = [imgUDLR,imgUD,imgUDLR;fliplr(img),img,fliplr(img)];
        rows = numel(img(:,1)); cols = numel(img(1,:)); % Image Dimensionality
        nSamp = ceil(sqrt(rowsˆ2+colsˆ2)/2); % Max # new sample points /2 for time saving estimation
% Solve Ray Intersection Coordinates
    offI = rows+1; offJ = cols+1;
    for i = 1:iF for j = 1:jF
        if rays.bool(i,j,iteration) ~= 0 % Determine if ray is completed -> 1 = active ray
            if iteration == 2 % Cell
                x1pro = j+offJ;
                x2pro = j+offJ+nSamp.*cos(rays.azim(i,j));
                xPro = [x1pro, x2pro];
                y1pro = i+offI;
                y2pro = i+offI-nSamp.*sin(rays.azim(i,j));
                yPro = [y1pro, y2pro];
                z1pro = imgTess(i+offI,j+offJ);
            else % Between 1 & 2
                x1pro = rays.xL(i,j,iteration-1);
                x2pro = rays.xL(i,j,iteration-1) + ... nSamp.*cos(rays.azim(i,j,iteration-1));
                xPro = [x1pro, x2pro];
                y1pro = rays.yL(i,j,iteration-1);
                y2pro = rays.yL(i,j,iteration-1) + ... nSamp.*sin(rays.azim(i,j,iteration-1));
                yPro = [y1pro, y2pro];
                z1pro = rays.zL(i,j,iteration-1);
            end
        a = improfile(imgTess,xPro,yPro,nSamp);
        a(isnan(a))=[];
        k = 2; % Reset Iterative imProfile Counter
        while k <= numel(a) && k ~= -1
            if (a(l) + k*tan(rays.elev(i,j))) < a(k)
                % Determine if ray interaction occurs
                x1 = k-1; x2 = k; x3 = k-1; x4 = k;
                % x points
                y1 = a(l) + (k-2)*tan(rays.elev(i,j));
                y2 = a(l) + (k-1)*tan(rays.elev(i,j));
                y3 = a(k-1); y4 = a(k); % y points
                xVec = [x1,x2,x3,x4]; yVec = [y1,y2,y3,y4];
                [Px, Py] = findLinesIntersect(xVec, yVec);
% Determine point of intersection
                delX = Px.*cos(rays.azim(i,j));
                delY = Px.*sin(rays.azim(i,j));
                delZ = Py - a(l);
                rays.xL(i,j,iteration) = x1pro + delX;
                % Shift x-coordinate (3 between 1 & 2)
                rays.yL(i,j,iteration) = y1pro + delY;
                % Shift y-coordinate
                rays.zL(i,j,iteration) = z1pro + delZ;
                % Shift z-coordinate
                error('solvetunnel is not an integer!');
            end
        end
    end
end
Listing 6.2: Calculate surface normals at all sampled points

```matlab
function [azimuths,elevations,rhos,vecx,vecy,vecz] = rayNormalsFun(img)
    %rayNormalsFun.m - Computes ray normals of input image
    if ~ismatrix(img)
        error('Input must be a matrix!');
    end
    img = double(img(1:964,:)); % Strip Metadata
    imR = numel(img(:,1)); imC = numel(img(1,:));
    % Compute x- and y- angular normals
    % Preallocations
    rx = zeros(size(img)-2); % -2 to eliminate first and last
    ry = rx;
    xAngs = rx;
    yAngs = rx;
    v3 = [rays.xL(i,j,iteration), ...
          rays.yL(i,j,iteration), ...
          rays.zL(i,j,iteration)];
    % Ray interaction position
    propVec = [-delX -delY -delZ];
    propVec = propVec./norm(propVec);
    nVec = findInterpolatedNormal(imgTess, v3);
    if sum(nVec) == -3
        rays.rayDepth(i,j) = iteration;
        rays.bool(i,j,iteration:end) = 0;
        k = -1; % Exit Loop Immediately
    else
        nVec = nVec./norm(nVec);
        % Determine Non-Integer Vec Normal
        [rays.azim(i,j,iteration), ...
         rays.elev(i,j,iteration), ...
         reflVec ] = reflRayFun2(nVec, ...
         propVec); % New Refl Vector
        rays.azim(i,j,iteration:end) = ...
        rays.azim(i,j,iteration);
        rays.elev(i,j,iteration:end) = ...
        rays.elev(i,j,iteration);
        rays.x(i,j,iteration:end) = reflVec(1);
        rays.y(i,j,iteration:end) = reflVec(2);
        rays.z(i,j,iteration:end) = reflVec(3);
        rays.power(i,j,iteration:end) = ...
        rays.power(i,j,iteration) + 1;
        % Minimize Ray Power
        rays.rayDepth(i,j) = iteration;
        k = -1; % Exit Loop Immediately
    end
    elseif k == numel(a) % Final case 'intersection
        rays.rayDepth(i,j) = iteration;
        rays.bool(i,j,iteration:end) = 0;
        % Raytrace Completed for i,j value
        k = -1; % Exit Loop Immediately
    else
        k = k+1;
    end
end
fprintf('Depth %d, Row %d completed!\n',iteration,i);
end
toc
save('rays')
```
azimuths = rx;
elevations = rx;
rhos = rx;
vecx = rx; vecy = rx; vecz = rx; % Final normal vector, all components

for j = 2:(imC-1)
    nArray = xSectionNormals(img(:,j));
    ry(:,j-1) = nArray(:,1);
    yAngs(:,j-1) = nArray(:,2);
end
for i = 2:(imR-1)
    nArray = xSectionNormals(img(i,:));
    rx(i-1,:) = nArray(:,1);
    xAngs(i-1,:) = nArray(:,2);
end
for i = 1:(imR-2)
    for j = 1:(imC-2)
        xx = rx(i,j).*cos(xAngs(i,j));
        xz = rx(i,j).*sin(xAngs(i,j));
        yy = ry(i,j).*cos(yAngs(i,j));
        yz = ry(i,j).*sin(yAngs(i,j));
        vx = [xx,0,xz]; vy = [0,yy,yz]; vz = (vx + vy);
        vecx(i,j) = vz(1);
        vecy(i,j) = vz(2);
        vecz(i,j) = vz(3);
        [gam,alph,rho] = cart2sph(vz(1),vz(2),vz(3));
        azimuths(i,j) = gam;
        elevations(i,j) = alph;
        rhos(i,j) = rho;
    end
end
end

Listing 6.3: Nested function call to calculate singular surface normal based on neighboring pixel intensities

function normArray = xSectionNormals(vec)
% Computes the cross-sectional angular normals to a vector input; normArray
% is an array of size [numel(vec), 2] correlating vector ends to each
% positional index, 1) Magnitude & 2) Angle, computed from (+) vector axis
vec = double(vec);
if ~isvector(vec) || ~isa(vec, 'double');
    error('Input must be a vector of double type!');
end
normArray = zeros(numel(vec)-2,2); % Eliminate Boundaries
for i = 2:(numel(vec)-1)
    ang1 = atan((vec(i)-vec(i-1))); % Angle from x-axis to (i-1)
    ang2 = atan((vec(i+1)-vec(i))); % Angle from x-axis to i+1
    if ang1 >= 0 && ang2 >= 0
        ang = (pi + ang1 - ang2)/2 + ang2;
    elseif ang1 >= 0 && ang2 < 0
        ang = pi + ang2 - (pi - ang1 + ang2)/2;
    elseif ang1 < 0 && ang2 >= 0
        ang = (pi - ang2 + ang1)/2 + ang2;
    elseif ang1 < 0 && ang2 < 0
        ang = (pi - ang2 + ang1)/2 + ang2;
    else
        error('Ray Angle Calculation Error');
    end
    r1 = sqrt(1^2+(vec(i-1)-vec(i))^2);
    r2 = sqrt(1^2+(vec(i)-vec(i+1))^2);
    r = (r1+r2)/2;
    normArray(i-1,1) = r;
Listing 6.4: Calculate initial reflected rayset array based on incident propagation vector

```matlab
function [ reflAzim, reflElev, rhos, reflX, reflY, reflZ ] = reflRayFun( vecx, vecy, vecz, propVec )
%
% reflRayFun.m - Determine reflected rayset given normal vector
%
% Components and input propagation vector
if sum(size(vecx) ~= size(vecy) & size(vecx) ~= size(vecz))
    error(['Matrices with normal vector components must be equal!'])
end
imR = numel(vecx(:,1)); imC = numel(vecx(1,:));
reflX = zeros(size(vecx)); reflY = reflX; reflZ = reflX;
reflAzim = reflX; reflElev = reflX; rhos = reflX;
for i = 1:imR
    for j = 1:imC
        nVec = [vecx(i,j), vecy(i,j), vecz(i,j)];
        reflVec = propVec - 2.*dot(propVec,nVec).*nVec;
        reflX(i,j) = reflVec(1);
        reflY(i,j) = reflVec(2);
        reflZ(i,j) = reflVec(3);
        [reflAzim(i,j),reflElev(i,j),rhos(i,j)] = cart2sph(reflX(i,j),reflY(i,j),reflZ(i,j));
    end
end
```

Listing 6.5: Determine successive ray-to-surface interaction

```matlab
function [ Px, Py ] = findLinesIntersect( x, y )
%
% Determines the intersection of two lines based on four sets of points,
% [x1,x2,x3,x4] and [y1,y2,y3,y4]
%
% Solve Inner Determinants
dtA = det([x(1),y(1);x(2),y(2)]);
dtA1 = det([x(1),1;x(2),1]);
dtB1 = det([y(1),1;y(2),1]);
dtC = det([x(3),y(3);x(4),y(4)]);
dtC1 = det([x(3),1;x(4),1]);
dtD1 = det([y(3),1;y(4),1]);
%
% Solve Outer Determinants
Pdenom = det([dtA,dtA1;dtC,dtC1]);
PxNum = det([dtA(dtA1,dtB1);dtC,dtC1]);
PyNum = det([dtA,dtA1;dtC,dtD1]);
%
% Division Stuff
Px = PxNum/Pdenom;
Py = PyNum/Pdenom;
end
```

Listing 6.6: Determine new normal vector at an interpolated surface point

```matlab
function normalVec = findInterpolatedNormal( imgTess, v3 )
%
% Find the normal vector of a non-integer index
% imgTess - Image to interpret
% v3 - original starting position of the new normal vector
cols = numel(imgTess(:,1)); rows = numel(imgTess(:,1));
```

140
if mod(v3(2),1) == 0 % Condition of ray occurs on boundary
    xVec = floor(v3(2))-1:ceil(v3(2))+2;
else
    xVec = floor(v3(2))-1:ceil(v3(2))+1;
end
if mod(v3(1),1) == 0 % Condition of ray occurs on boundary
    yVec = floor(v3(1))-1:ceil(v3(1))+2;
else
    yVec = floor(v3(1))-1:ceil(v3(1))+1;
end
if (min(xVec) < 1 || max(xVec) > cols)|| (min(yVec) < 1 || max(yVec) > rows)
    normalVec = [-1 -1 -1];
else
    normCond = imgTess(yVec,xVec); % Condensed Normalization Array
    [˜, ˜, ˜, vecxP, vecyP, veczP] = rayNormalsFun( normCond );
    % If non-integer position
    VqX = interp2(vecxP, 1+mod(v3(1),1), 1+mod(v3(2),1));
    % Find weighted normal, x
    VqY = interp2(vecyP, 1+mod(v3(1),1), 1+mod(v3(2),1));
    % Find weighted normal, y
    VqZ = interp2(veczP, 1+mod(v3(1),1), 1+mod(v3(2),1));
    % Find weighted normal, z
    normalVec = [VqX VqY VqZ];
    normC = norm(normalVec);
    normalVec = normalVec./normC;
end
end

Listing 6.7: Calculate singular reflected ray vector based on incident propagation ray and interpolated surface normal vector

function [reflAzim,reflElev,rho,reflVec] = reflRayFun2(normVec,propVec)
% Reflects two vectors
if sum(size(normVec) ~= size(propVec) & isvector(normVec))
    error('Function only accepts vectors, not arrays!')
end
reflVec = normVec - 2.*dot(normVec,propVec).*propVec; % New Refl Vector
[reflAzim,reflElev,rho] = cart2sph(reflVec(1),reflVec(2),reflVec(3));
end

C++ Runtime, Supported with MATLAB Coder Port

Listing 6.8: Main raytracing runtime via C++ port: Header File

#ifndef MAIN_H
#define MAIN_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"
#include "rayTraceC_types.h"

// Function Declarations
extern int main(int argc, const char * const argv[]);
#endif
Listing 6.9: Main raytracing runtime via C++ port: C++ File

```c
#include "rt_nonfinite.h"
#include "main.h"
#include "rayTraceC.h"
#include "rayTraceC_terminate.h"
#include "rayTraceC_emxAPI.h"
#include "rayTraceC_initialize.h"
#include <stdio.h>
#include "libmwjpegreader.h"

// Function Declarations
static void main_rayTraceC();

// Function Definitions
// Arguments: void
// Return Type: void
static void main_rayTraceC()
{
    struct0_T rays;
    emxInit_struct0_T(&rays);
    // Call the entry-point 'rayTraceC'.
    rayTraceC(&rays);
    emxDestroy_struct0_T(rays);
}

// Arguments: int argc
// Return Type: int
int main(int, const char * const [])
{
    // Initialize the application.
    // You do not need to do this more than one time.
    rayTraceC_initialize();
    // Invoke the entry-point functions.
    // You can call entry-point functions multiple times.
    main_rayTraceC();
    // Terminate the application.
    // You do not need to do this more than one time.
    rayTraceC_terminate();
    return 0;
}
```

Listing 6.10: Ray Normals Calculation: Header File

```c
#ifndef RAYNORMALSFUN_H
#define RAYNORMALSFUN_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"
#include "rayTraceC_types.h"

// Function Declarations
extern void rayNormalsFun(const emxArray_real_T *img, emxArray_real_T *azimuths,
                           emxArray_real_T *elevations, emxArray_real_T *rhos, emxArray_real_T *vecx,
                           emxArray_real_T *vecy, emxArray_real_T *vecz);
#endif
```

Listing 6.11: Ray Normals Calculation: C++ File

```c
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"
#include "rayTraceC_types.h"

// Function Declarations
extern void rayNormalsFun(const emxArray_real_T *img, emxArray_real_T *azimuths,
                           emxArray_real_T *elevations, emxArray_real_T *rhos, emxArray_real_T *vecx,
                           emxArray_real_T *vecy, emxArray_real_T *vecz);
```
#include "rt_nonfinite.h"
#include "rayTraceC.h"
#include "xSectionNormals.h"
#include "cart2sph.h"
#include "rayTraceC_emxutil.h"
#include <stdio.h>

void rayNormalsFun(const emxArray_real_T *img, emxArray_real_T *azimuths,
emxArray_real_T *elevations, emxArray_real_T *rhos,
emxArray_real_T *vecx, emxArray_real_T *vecy, emxArray_real_T
*vecz)
{
    int i4;
    short iv4[2];
    int i5;
    emxArray_real_T *rx;
    int loop_ub;
    emxArray_real_T *ry;
    emxArray_real_T *xAngs;
    emxArray_real_T *yAngs;
    int j;
    int exitg1;
    double nArray[1924];
    emxArray_real_T *r0;
    double nArray_data[7676];
    int tmp_data[3838];
    int i;
    double img_data[3840];
    int img_size[2];
    double b_rx[3];
    emxArray_real_T b_img_data;
    double dv1[3];
    double vz[3];
    int nArray_size_idx_0;
    double gam;
    double alph;
    double rho;
    double b_nArray_data[3838];

    // img = double(img(1:964,:)); % Strip Metadata
    i4 = img->size[1];

    // % Compute x- and y- angular normals
    // Preallocations
    for (i5 = 0; i5 < 2; i5++) {
        iv4[i5] = (short)((short)img->size[i5] - 2);
    }
    emxInit_real_T(&rx, 2);
    i5 = rx->size[0] * rx->size[1];
    rx->size[0] = 962;
    rx->size[1] = iv4[1];
    emxEnsureCapacity((emxArray__common *)rx, i5, (int)sizeof(double));
    loop_ub = 962 * iv4[1];
    for (i5 = 0; i5 < loop_ub; i5++) {
        rx->data[i5] = 0.0;
    }
    emxInit_real_T(&ry, 2);

    // - 2 to eliminate first and last
    i5 = ry->size[0] * ry->size[1];
    ry->size[0] = 962;
    ry->size[1] = iv4[1];
```c
emxEnsureCapacity((emxArray__common *)ry, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  ry->data[i5] = 0.0;
}
emxInit_real_T(&xAngs, 2);
i5 = xAngs->size[0] * xAngs->size[1];
xAngs->size[0] = 962;
xAngs->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)xAngs, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  xAngs->data[i5] = 0.0;
}
emxInit_real_T(&yAngs, 2);
i5 = yAngs->size[0] * yAngs->size[1];
yAngs->size[0] = 962;
yAngs->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)yAngs, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  yAngs->data[i5] = 0.0;
}
i5 = azimuths->size[0] * azimuths->size[1];
azimuths->size[0] = 962;
azimuths->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)azimuths, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  azimuths->data[i5] = 0.0;
}
i5 = elevations->size[0] * elevations->size[1];
elevations->size[0] = 962;
elevations->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)elevations, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  elevations->data[i5] = 0.0;
}
i5 = rhos->size[0] * rhos->size[1];
rhos->size[0] = 962;
rhos->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)rhos, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  rhos->data[i5] = 0.0;
}
i5 = vecx->size[0] * vecx->size[1];
vecx->size[0] = 962;
vecx->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)vecx, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
  vecx->data[i5] = 0.0;
}
i5 = vecy->size[0] * vecy->size[1];
vecy->size[0] = 962;
vecy->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)vecy, i5, (int)sizeof(double));
```

loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
    vecy->data[i5] = 0.0;
}

i5 = vecz->size[0] * vecz->size[1];
vecz->size[0] = 962;
vecz->size[1] = iv4[1];
emxEnsureCapacity((emxArray__common *)vecz, i5, (int)sizeof(double));
loop_ub = 962 * iv4[1];
for (i5 = 0; i5 < loop_ub; i5++) {
    vecz->data[i5] = 0.0;
}

// Final normal vector, all components
i5 = 0;
do {
    exitg1 = 0;
    i5 = img->size[1];
    if (j <= i5 - 3) {
        xSectionNormals((double **)img->data[img->size[0] * (1 + j)],
                        nArray);
        for (i5 = 0; i5 < 962; i5++) {
            ry->data[i5 + ry->size[0] * j] = nArray[i5];
        }
    }
    for (i5 = 0; i5 < 962; i5++) {
        yAngs->data[i5 + yAngs->size[0] * j] = nArray[962 + i5];
    }
    j++;
    } else {
    exitg1 = 1;
    }
}while (exitg1 == 0);
emxInit_real_T(&r0, 2);
for (i = 0; i < 962; i++) {
    loop_ub = img->size[1];
    img_size[0] = 1;
    img_size[1] = loop_ub;
    for (i5 = 0; i5 < loop_ub; i5++) {
        img_data[img_size[0] * i5] = img->data[(i + img->size[0] * i5) + 1];
    }
}

b_img_data.data = (double *)&img_data;
b_img_data.size = (int *)&img_size;
b_img_data.allocatedSize = 3840;
b_img_data.numDimensions = 2;
b_img_data.canFreeData = false;
b_xSectionNormals(&b_img_data, r0);
Array_size_idx_0 = r0->size[0];
loop_ub = r0->size[0] * r0->size[1];
for (i5 = 0; i5 < loop_ub; i5++) {
    nArray_data[i5] = r0->data[i5];
}

loop_ub = rx->size[1];
for (i5 = 0; i5 < loop_ub; i5++) {
    tmp_data[i5] = i5;
}

for (i5 = 0; i5 < nArray_size_idx_0; i5++) {
    b_nArray_data[i5] = nArray_data[i5];
}
for (i5 = 0; i5 < loop_ub; i5++) {
    rx->data[i + rx->size[0] * tmp_data[i5]] = b_nArray_data[i5];
}

loop_ub = xAngs->size[1];
for (i5 = 0; i5 < loop_ub; i5++) {
    tmp_data[i5] = i5;
}
for (i5 = 0; i5 < nArray_size_idx_0; i5++) {
    b_nArray_data[i5] = nArray_data[i5 + nArray_size_idx_0];
}
for (i5 = 0; i5 < loop_ub; i5++) {
    xAngs->data[i + xAngs->size[0] * tmp_data[i5]] = b_nArray_data[i5];
}
emxFree_real_T(&r0);
for (i = 0; i < 962; i++) {
    for (j = 0; j <= i4 - 3; j++) {
        b_rx[0] = rx->data[i + rx->size[0] * j] * cos(xAngs->data[i + xAngs->size[0] * j]);
        b_rx[1] = 0.0;
        b_rx[2] = rx->data[i + rx->size[0] * j] * sin(xAngs->data[i + xAngs->size[0] * j]);
        dv1[0] = 0.0;
        dv1[1] = ry->data[i + ry->size[0] * j] * cos(yAngs->data[i + yAngs->size[0] * j]);
        for (i5 = 0; i5 < 3; i5++) {
            vz[i5] = b_rx[i5] + dv1[i5];
        }
        vecx->data[i + vecx->size[0] * j] = vz[0];
        vecy->data[i + vecy->size[0] * j] = vz[1];
        vecz->data[i + vecz->size[0] * j] = vz[2];
        cart2sph(vz[0], vz[1], vz[2], &gam, &alph, &rho);
        // ~ is radii, replace if necessary
        azimuths->data[i + azimuths->size[0] * j] = gam;
        elevations->data[i + elevations->size[0] * j] = alph;
        rhos->data[i + rhos->size[0] * j] = rho;
        // gamma(i,j) = atan(vz(2)/vz(1));
        // alpha(i,j) = atan(vz(3)/sqrt(vz(1)^2+vz(2)^2));
    }
}
emxFree_real_T(&yAngs);
emxFree_real_T(&xAngs);
emxFree_real_T(&ry);
emxFree_real_T(&rx);
}

Listing 6.12: Calculate Cross-Section Normals: Header File

#ifndef XSECTIONNORMALS_H
#define XSECTIONNORMALS_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>

#endif XSECTIONNORMALS_H
Listing 6.13: Calculate Cross-Section Normals: C++ File
ang = ((3.1415926535897931 - ang2) + ang1) / 2.0 + ang2;

// Correct
}

a = vec->data[i] - vec->data[1 + i];
b_a = vec->data[1 + i] - vec->data[2 + i];
normArray->data[i] = (sqrt(1.0 + a * a) + sqrt(1.0 + b_a * b_a)) / 2.0;
normArray->data[i + normArray->size[0]] = ang;
}

// Computes the cross-sectional angular normals to an input vector
// normArray is an array of size [numel(vec), 2] correlating vector ends to
// each positional index, 1) Magnitude & 2) Angle, computed from positive vector axis.
// Arguments : const double vec[964]
// Return Type : void

void xSectionNormals(const double vec[964], double normArray[1924])
{
    int i;
    double ang1;
    double ang2;
    double ang;
    double a;
    double b_a;

    // Eliminate Boundaries
    for (i = 0; i < 962; i++) {
        ang1 = atan(vec[i + 1] - vec[i]);
        // Angle from x-axis to (i-1)i
        ang2 = atan(vec[2 + i] - vec[i + 1]);
        // Angle from x-axis to i(i+1)
        // normArray(i,1) = ang1; normArray(i,2) = ang2;

        if ((ang1 >= 0.0) && (ang2 >= 0.0)) {
            ang = ((3.1415926535897931 + ang1) - ang2) / 2.0 + ang2;
            // Correct
        } else if ((ang1 >= 0.0) && (ang2 < 0.0)) {
            ang = (3.1415926535897931 + ang2) - ((3.1415926535897931 - ang1) + ang2) / 2.0;
            // Correct
        } else if ((ang1 < 0.0) && (ang2 >= 0.0)) {
            ang = ((3.1415926535897931 - ang2) + ang1) / 2.0 + ang2;
            // Correct
        } else {
            ang = ((3.1415926535897931 - ang2) + ang1) / 2.0 + ang2;
            // Correct
        }
    }

    a = vec[i] - vec[i + 1];
b_a = vec[1 + i] - vec[2 + i];
normArray[i] = (sqrt(1.0 + a * a) + sqrt(1.0 + b_a * b_a)) / 2.0;
normArray[962 + i] = ang;
}
Listing 6.14: Calculate Specular Reflection Array: Header File

```c
#ifndef REFLRAYFUN_H
#define REFLRAYFUN_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"
#include "rayTraceC_types.h"

// Function Declarations
extern void reflRayFun(const emxArray_real_T *vecx, const emxArray_real_T *vecy, const emxArray_real_T *vecz, emxArray_real_T *reflAzim, emxArray_real_T *reflElev, emxArray_real_T *rhos, emxArray_real_T *reflX, emxArray_real_T *reflY, emxArray_real_T *reflZ);
#endif
```

Listing 6.15: Calculate Specular Reflection Array: C++ File

```cpp
#include "rt_nonfinite.h"
#include "rayTraceC.h"
#include "reflRayFun.h"
#include "cart2sph.h"
#include "dot.h"
#include "rayTraceC_emxutil.h"
#include <stdio.h>
#include "libmwjpegreader.h"

void reflRayFun(const emxArray_real_T *vecx, const emxArray_real_T *vecy, const emxArray_real_T *vecz, emxArray_real_T *reflAzim, emxArray_real_T *reflElev, emxArray_real_T *rhos, emxArray_real_T *reflX, emxArray_real_T *reflY, emxArray_real_T *reflZ)
{
    int i7;
    short iv5[2];
    int i8;
    int loop_ub;
    int i;
    int j;
    double nVec[3];
    double d0;
    static const double propVec[3] = { 0.0, 0.0, -1.0 };

    i7 = vecx->size[1];
    for (i8 = 0; i8 < 2; i8++) {
        iv5[i8] = (short)vecx->size[i8];
    }

    i8 = reflX->size[0] * reflX->size[1];
    reflX->size[0] = 962;
    reflX->size[1] = iv5[1];
    emxEnsureCapacity((emxArray__common *)reflX, i8, (int)sizeof(double));
    loop_ub = 962 * iv5[1];
    for (i8 = 0; i8 < loop_ub; i8++) {
        reflX->data[i8] = 0.0;
    }

    i8 = reflY->size[0] * reflY->size[1];
    reflY->size[0] = 962;
```

149
reflY->size[1] = iv5[1];
emxEnsureCapacity((emxArray__common *)reflY, i8, (int)sizeof(double));
loop_ub = 962 * iv5[1];
for (i8 = 0; i8 < loop_ub; i8++) {
    reflY->data[i8] = 0.0;
}

i8 = reflZ->size[0] * reflZ->size[1];
reflZ->size[0] = 962;
reflZ->size[1] = iv5[1];
emxEnsureCapacity((emxArray__common *)reflZ, i8, (int)sizeof(double));
loop_ub = 962 * iv5[1];
for (i8 = 0; i8 < loop_ub; i8++) {
    reflZ->data[i8] = 0.0;
}

i8 = reflAzim->size[0] * reflAzim->size[1];
reflAzim->size[0] = 962;
reflAzim->size[1] = iv5[1];
emxEnsureCapacity((emxArray__common *)reflAzim, i8, (int)sizeof(double));
loop_ub = 962 * iv5[1];
for (i8 = 0; i8 < loop_ub; i8++) {
    reflAzim->data[i8] = 0.0;
}

i8 = reflElev->size[0] * reflElev->size[1];
reflElev->size[0] = 962;
reflElev->size[1] = iv5[1];
emxEnsureCapacity((emxArray__common *)reflElev, i8, (int)sizeof(double));
loop_ub = 962 * iv5[1];
for (i8 = 0; i8 < loop_ub; i8++) {
    reflElev->data[i8] = 0.0;
}

i8 = rhos->size[0] * rhos->size[1];
rhos->size[0] = 962;
rhos->size[1] = iv5[1];
emxEnsureCapacity((emxArray__common *)rhos, i8, (int)sizeof(double));
loop_ub = 962 * iv5[1];
for (i8 = 0; i8 < loop_ub; i8++) {
    rhos->data[i8] = 0.0;
}

for (i = 0; i < 962; i++) {
    for (j = 0; j < i7; j++) {
        nVec[0] = vecx->data[i + vecx->size[0] * j];
        nVec[1] = vecy->data[i + vecy->size[0] * j];
        nVec[2] = vecz->data[i + vecz->size[0] * j];
        d0 = 2.0 * dot(propVec, nVec);
        for (i8 = 0; i8 < 3; i8++) {
            nVec[i8] = propVec[i8] - d0 * nVec[i8];
        }
        // Should be nVec = 2.*dot(nVec,propVec).*propVec ?
        reflX->data[i + reflX->size[0] * j] = nVec[0];
        reflY->data[i + reflY->size[0] * j] = nVec[1];
        reflZ->data[i + reflZ->size[0] * j] = nVec[2];
        cart2sph(reflX->data[i + reflX->size[0] * j], reflY->data[i + reflY->size
[0] * j], reflZ->data[i + reflZ->size[0] * j], &reflAzim->data[i + reflAzim->size[0] * j], &reflElev->data[i + reflElev->size[0] * j], &rhos->data[i + rhos->size[0] * j]);
    }
}
Listing 6.16: Calculate Specular Reflection Vector: Header File

```c
#ifndef REFLRAYFUN2_H
#define REFLRAYFUN2_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"
#include "rayTraceC_types.h"

// Function Declarations
extern void reflRayFun2(const double normVec[3], const double propVec[3], double *
reflAzim, double *reflElev, double *rho, double reflVec[3]);
#endif
```

Listing 6.17: Calculate Specular Reflection Vector: C++ File

```cpp
#include "rt_nonfinite.h"
#include "rayTraceC.h"
#include "reflRayFun2.h"
#include "rayTraceC_rtwutil.h"
#include <stdio.h>
#include "libmwjpegreader.h"

void reflRayFun2(const double normVec[3], const double propVec[3], double *
reflAzim, double *reflElev, double *rho, double reflVec[3])
{

double hypotxy;
int k;
hypotxy = 0.0;
for (k = 0; k < 3; k++) {
    hypotxy += normVec[k] * propVec[k];
}

hypotxy *= 2.0;
for (k = 0; k < 3; k++) {
    reflVec[k] = normVec[k] - hypotxy * propVec[k];
}

// New Refl Vector
hypotxy = rt_hypotd_snf(reflVec[0], reflVec[1]);
rho = rt_hypotd_snf(hypotxy, reflVec[2]);
*reflElev = rt_atan2d_snf(reflVec[2], hypotxy);
*reflAzim = rt_atan2d_snf(reflVec[1], reflVec[0]);
}
```

Listing 6.18: Calculate Specular Reflection Vector: Header File

```c
#ifndef FINDLINESINTERSECT_H
#define FINDLINESINTERSECT_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"

```
Listing 6.19: Calculate Specular Reflection Vector: C++ File

```cpp
#include "rt_nonfinite.h"
#include "rayTraceC.h"
#include "findLinesIntersect.h"
#include "det.h"
#include <stdio.h>
#include "libmwjpegreader.h"

void findLinesIntersect(const double x[4], const double y[4], double *Px, double *Py)
{
    double b_x[4];
    double dtA;
    double c_x[4];
    double dtA1;
    double b_y[4];
    double dtB1;
    double d_x[4];
    double dtC;
    double e_x[4];
    double dtC1;
    double c_y[4];
    double dtD1;
    double b_dtA[4];
    double Pdenom;
    double b_dtA[4];
    double c_dtA[4];

    // Solve Inner Determinants
    b_x[0] = x[0];
    b_x[2] = y[0];
    b_x[1] = x[1];
    b_x[3] = y[1];
    dtA = det(b_x);
    c_x[0] = x[0];
    c_x[2] = 1.0;
    c_x[1] = x[1];
    c_x[3] = 1.0;
    dtA1 = det(c_x);
    b_y[0] = y[0];
    b_y[2] = 1.0;
    b_y[1] = y[1];
    b_y[3] = 1.0;
    dtB1 = det(b_y);
    d_x[0] = x[2];
    d_x[2] = y[2];
    d_x[1] = x[3];
    d_x[3] = y[3];
    dtC = det(d_x);
    e_x[0] = x[2];
    e_x[2] = 1.0;
    e_x[1] = x[3];
    e_x[3] = 1.0;
    dtC1 = det(e_x);
    c_y[0] = y[2];
    c_y[2] = 1.0;
    c_y[1] = 0.0;
}
```
\texttt{c_y[1] = y[3];}
\texttt{c_y[3] = 1.0;}
\texttt{dtD1 = det(c_y);}

// Solve Outer Determinants
\texttt{b_dtA1[0] = dtA;}
\texttt{b_dtA1[1] = dtC1;}
\texttt{b_dtA1[3] = dtD1;}
\texttt{Pdenom = det(b_dtA1);}
\texttt{b_dtA[0] = dtA;}
\texttt{b_dtA[1] = dtC;}
\texttt{b_dtA[3] = dtD1;}
\texttt{c_dtA[0] = dtA;}
\texttt{c_dtA[1] = dtC;}
\texttt{c_dtA[3] = dtD1;}

// Division Stuff
\texttt{*Px = det(b_dtA) / Pdenom;}
\texttt{*Py = det(c_dtA) / Pdenom;}

\textbf{Listing 6.20: Calculate Interpolated Normal: Header File}

```
#ifndef FINDINTERPOLATEDNORMAL_H
#define FINDINTERPOLATEDNORMAL_H

// Include Files
#include <math.h>
#include <stddef.h>
#include <stdlib.h>
#include <string.h>
#include "rt_defines.h"
#include "rt_nonfinite.h"
#include "rtwtypes.h"
#include "rayTraceC_types.h"

// Function Declarations
extern void findInterpolatedNormal(const emxArray_real_T * imgTess, const double v3[3], double normalVec[3]);

#endif
```

\textbf{Listing 6.21: Calculate Interpolated Normal: C++ File}

```
#include "rt_nonfinite.h"
#include "rayTraceC.h"
#include "findInterpolatedNormal.h"
#include "rayTraceC_emxutil.h"
#include "xSectionNormals.h"
#include "norm.h"
#include "interp2.h"
#include "mod.h"
#include <stdio.h>
#include "libmwjpegreader.h"

void findInterpolatedNormal(const emxArray_real_T * imgTess, const double v3[3], double normalVec[3])
{
    emxArray_real_T * xVec;
    double x;
    double b_x;
    ```
double ndbl;
double anew;
int n;
double apnd;
double cdiff;
int i9;
double absa;
double absb;
emxArray_real_T *yVec;
int nm1d2;
int k;
emxArray_real_T *normCond;
emxArray_real_T *b_yVec;
int i10;
emxArray_real_T *b_imgTess;
int iv6[2];
emxArray_real_T *rx;
emxArray_real_T *ry;
emxArray_real_T *xAngs;
emxArray_real_T *yAngs;
emxArray_real_T *vecxP;
emxArray_real_T *vecyP;
emxArray_real_T *veczP;
int j;
emxArray_real_T *nArray;
emxArray_real_T *c_yVec;
emxArray_real_T *b_xVec;
emxArray_real_T *c_xVec;
emxArray_real_T *d_xVec;
emxArray_real_T *e_xVec;
emxArray_real_T *f_xVec;
emxArray_real_T *g_xVec;
emxArray_real_T *h_xVec;
emxArray_real_T *i_xVec;
emxArray_real_T *j_xVec;
emxArray_real_T *k_xVec;
int exitg3;
int i;
emxArray_int32_T *r1;
emxArray_real_T *b_normCond;
emxArray_real_T *l_xVec;
emxArray_real_T *d_yVec;
emxArray_real_T *b_nArray;
emxArray_real_T *c_nArray;
emxArray_real_T *m_xVec;
int exitg1;
double ang1;
emxArray_real_T *b_rx[3];
double ang2;
double dv2[3];
double ang;
double vz[3];
double VqX;
double VqY;
double Vq2;
emxInit_real_T(&xVec, 2);
if (b_mod(v3[1]) == 0.0) {
x = floor(v3[1]);
b_x = ceil(v3[1]);
ndbl = floor(v3[1]) - 1.0;
anew = ceil(v3[1]) + 2.0;
if (rtIsNaN(ndbl) || rtIsNaN(anew)) {
n = 1;
anew = rtNaN;
apnd = b_x + 2.0;
```c
} else if (b_x + 2.0 < x - 1.0) {
    n = 0;
    anew = x - 1.0;
    apnd = b_x + 2.0;
} else if (rtIsInf(ndbl) || rtIsInf(anew)) {
    n = 1;
    anew = rtNaN;
    apnd = b_x + 2.0;
} else {
    anew = x - 1.0;
    ndbl = floor(((b_x + 2.0) - (x - 1.0)) + 0.5);
    apnd = (x - 1.0) + ndbl;
    cdiff = apnd - (b_x + 2.0);
    absa = fabs(x - 1.0);
    absb = fabs(b_x + 2.0);
    if ((absa <= absb) || rtIsNaN(absb)) {
        absb = absa;
    }
    if (fabs(cdiff) < 4.4408920985006262E-16 * absb) {
        ndbl++;
        apnd = b_x + 2.0;
    } else if (cdiff > 0.0) {
        apnd = (x - 1.0) + (ndbl - 1.0);
    } else {
        ndbl++;
    }
    if (ndbl >= 0.0) {
        n = (int)ndbl;
    } else {
        n = 0;
    }
}

i9 = xVec->size[0] * xVec->size[1];
xVec->size[0] = 1;
xVec->size[1] = n;
emxEnsureCapacity((emxArray__common *)xVec, i9, (int)sizeof(double));
if (n > 0) {
    xVec->data[0] = anew;
    if (n > 1) {
        xVec->data[n - 1] = apnd;
        nm1d2 = (n - 1) / 2;
        for (k = 1; k < nm1d2; k++) {
            xVec->data[k] = anew + (double)k;
            xVec->data[(n - k) - 1] = apnd - (double)k;
        }
        if (nm1d2 << 1 == n - 1) {
            xVec->data[nm1d2] = (anew + apnd) / 2.0;
        } else {
            xVec->data[nm1d2] = anew + (double)nm1d2;
            xVec->data[nm1d2 + 1] = apnd - (double)nm1d2;
        }
    }
}
}
else if (v3[1]) {
    x = floor(v3[1]);
    b_x = ceil(v3[1]);
    ndbl = floor(v3[1]) - 1.0;
    anew = ceil(v3[1]) + 1.0;
    if (rtIsNaN(ndbl) || rtIsNaN(anew)) {
        n = 1;
        anew = rtNaN;
        apnd = b_x + 1.0;
    }
```
else if (b_x + 1.0 < x - 1.0) {
    n = 0;
    anew = x - 1.0;
    apnd = b_x + 1.0;
} else if (rtIsInf(ndbl) || rtIsInf(anew)) {
    n = 1;
    anew = rtNaN;
    apnd = b_x + 1.0;
} else {
    anew = x - 1.0;
    ndbl = floor(((b_x + 1.0) - (x - 1.0)) + 0.5);
    apnd = (x - 1.0) + ndbl;
    cdiff = apnd - (b_x + 1.0);
    absa = fabs(x - 1.0);
    absb = fabs(b_x + 1.0);
    if ((absa >= absb) || rtIsNaN(absb)) {
        absb = absa;
    }
    if (fabs(cdiff) < 4.4408920985006262E-16 * absb) {
        ndbl++;
        apnd = b_x + 1.0;
    } else if (cdiff > 0.0) {
        apnd = (x - 1.0) + (ndbl - 1.0);
    } else {
        ndbl++;
    }
    if (ndbl >= 0.0) {
        n = (int)ndbl;
    } else {
        n = 0;
    }
}

i9 = xVec->size[0] * xVec->size[1];
xVec->size[0] = 1;
xVec->size[1] = n;
emxEnsureCapacity((emxArray__common *)xVec, i9, (int)sizeof(double));
if (n > 0) {
    xVec->data[0] = anew;
    if (n > 1) {
        xVec->data[n - 1] = apnd;
        nm1d2 = (n - 1) / 2;
        for (k = 1; k < nm1d2; k++) {
            xVec->data[k] = anew + (double)k;
            xVec->data[(n - k) - 1] = apnd - (double)k;
        }
        if (nm1d2 <= 1 == n - 1) {
            xVec->data[nm1d2] = (anew + apnd) / 2.0;
        } else {
            xVec->data[nm1d2] = anew + (double)nm1d2;
            xVec->data[nm1d2 + 1] = apnd - (double)nm1d2;
        }
    }
}

emxInit_real_T(&yVec, 2);
if (b_mod(v3[0]) == 0.0) {
    x = floor(v3[0]);
    b_x = ceil(v3[0]);
    ndbl = floor(v3[0]) - 1.0;
    anew = ceil(v3[0]) + 2.0;
    if (rtIsNaN(ndbl) || rtIsNaN(anew)) {
n = 1;
anew = rtNaN;
apnd = b_x + 2.0;
} else if (b_x + 2.0 < x - 1.0) {
  n = 0;
anew = x - 1.0;
apnd = b_x + 2.0;
} else if (rtIsInf(ndbl) || rtIsInf(anew)) {
  n = 1;
anew = rtNaN;
apnd = b_x + 2.0;
} else {
  anew = x - 1.0;
  ndbl = floor(((b_x + 2.0) - (x - 1.0)) + 0.5);
apnd = (x - 1.0) + ndbl;
cdiff = apnd - (b_x + 2.0);
absa = fabs(x - 1.0);
absb = fabs(b_x + 2.0);
if ((absa >= absb) || rtIsNaN(absb)) {
  absb = absa;
}
  
  if (fabs(cdiff) < 4.4408920985006262E-16 * absb) {
    ndbl++;
apnd = b_x + 2.0;
  } else if (cdiff > 0.0) {
apnd = (x - 1.0) + (ndbl - 1.0);
  } else {
    ndbl++;
  }

  if (ndbl >= 0.0) {
    n = (int)ndbl;
  } else {
    n = 0;
  }
}

i9 = yVec->size[0] * yVec->size[1];
yVec->size[0] = 1;
yVec->size[1] = n;
emxEnsureCapacity((emxArray__common *)yVec, i9, (int)sizeof(double));
if (n > 0) {
yVec->data[0] = anew;
  if (n > 1) {
    yVec->data[n - 1] = apnd;
nmld2 = (n - 1) / 2;
    for (k = 1; k < nmld2; k++) {
yVec->data[k] = anew + (double)k;
yVec->data[(n - k) - 1] = apnd - (double)k;
  }
  }

  if (nmld2 <= 1 && n - 1) {
yVec->data[nmld2] = (anew + apnd) / 2.0;
  } else {
yVec->data[nmld2] = anew + (double)nmld2;
yVec->data[nmld2 + 1] = apnd - (double)nmld2;
  }
}

} else {
x = floor(v3[0]);
b_x = ceil(v3[0]);
ndbl = floor(v3[0]) - 1.0;
anew = ceil(v3[0]) + 1.0;
if (rtIsNaN(ndbl) || rtIsNaN(anew)) {
n = 1;
anew = rtNaN;
apnd = b_x + 1.0;
} else if (b_x + 1.0 < x - 1.0) {
    n = 0;
anew = x - 1.0;
apnd = b_x + 1.0;
} else if (rtIsInf(ndbl) || rtIsInf(anew)) {
    n = 1;
anew = rtNaN;
apnd = b_x + 1.0;
} else {
    anew = x - 1.0;
dbl = floor(((b_x + 1.0) - (x - 1.0)) + 0.5);
apnd = (x - 1.0) + ndbl;
cdiff = apnd - (b_x + 1.0);
absa = fabs(x - 1.0);
absb = fabs(b_x + 1.0);
if ((absa >= absb) || rtIsNaN(absb)) {
    absb = absa;
}

if (fabs(cdiff) < 4.4408920985006262E-16 * absb) {
    ndbl++;
apnd = b_x + 1.0;
} else if (cdiff > 0.0) {
    apnd = (x - 1.0) + (ndbl - 1.0);
} else {
    ndbl++;
}

if (ndbl >= 0.0) {
    n = (int)ndbl;
} else {
    n = 0;
}

i9 = yVec->size[0] * yVec->size[1];
yVec->size[0] = 1;
yVec->size[1] = n;
emxEnsureCapacity((emxArray__common *) yVec, i9, (int)sizeof(double));
if (n > 0) {
    yVec->data[0] = anew;
    if (n > 1) {
        yVec->data[n - 1] = apnd;
        mld2 = (n - 1) / 2;
        for (k = 1; k < mld2; k++) {
            yVec->data[k] = anew + (double)k;
yVec->data[(n - k) - 1] = apnd - (double)k;
        }
        if (mld2 << 1 == n - 1) {
            yVec->data[mld2] = (anew + apnd) / 2.0;
        } else {
            yVec->data[mld2] = anew + (double)mld2;
yVec->data[mld2 + 1] = apnd - (double)mld2;
        }
    }
}

emxInit_real_T(&normCond, 2);
i9 = normCond->size[0] * normCond->size[1];
normCond->size[0] = xVec->size[1];
normCond->size[1] = yVec->size[1];
emxEnsureCapacity((emxArray__common *)normCond, i9, (int)sizeof(double));

k = yVec->size[1];
for (i9 = 0; i9 < k; i9++) {
  nm1d2 = xVec->size[1];
  for (i10 = 0; i10 < nm1d2; i10++) {
    normCond->data[i10 + normCond->size[0] * i9] = imgTess->data[((int)xVec->size[0] * i10) + imgTess->size[0] * ((int)yVec->data[yVec->size[0] * i9] - 1)] - 1;
  }
}

emxInit_real_T2(&b_yVec, 1);

// Condensed Normalization Array
// rayNormalsFun.m - Computes ray normals of input image
// img = double(img(1:964,:)); % Strip Metadata
i9 = b_yVec->size[0];
b_yVec->size[0] = yVec->size[1];
emxEnsureCapacity((emxArray__common *)b_yVec, i9, (int)sizeof(double));
k = yVec->size[1];
for (i9 = 0; i9 < k; i9++) {
  b_yVec->data[i9] = yVec->data[yVec->size[0] * i9];
}

emxInit_real_T(&b_imgTess, 2);
i9 = b_yVec->size[0];

// % Compute x- and y- angular normals
// Preallocations
i10 = b_imgTess->size[0] + b_imgTess->size[1];
b_imgTess->size[0] = xVec->size[1];
b_imgTess->size[1] = yVec->size[1];
emxEnsureCapacity((emxArray__common *)b_imgTess, i10, (int)sizeof(double));
k = yVec->size[1];
emxFree_real_T(&b_yVec);
for (i10 = 0; i10 < k; i10++) {
  nm1d2 = xVec->size[1];
  for (n = 0; n < nm1d2; n++) {
    b_imgTess->data[n + b_imgTess->size[0] * i10] = imgTess->data[((int)xVec->size[0] * n) + imgTess->size[0] * ((int)yVec->data[yVec->size[0] * i10] - 1)] - 1;
  }
}

for (i10 = 0; i10 < 2; i10++) {
  iv6[i10] = b_imgTess->size[i10] - 2;
}
emxFree_real_T(&b_imgTess);
emxInit_real_T(&rx, 2);
i10 = rx->size[0] + rx->size[1];
rx->size[0] = iv6[0];
rx->size[1] = iv6[1];
emxEnsureCapacity((emxArray__common *)rx, i10, (int)sizeof(double));
k = iv6[0] * iv6[1];
for (i10 = 0; i10 < k; i10++) {
  rx->data[i10] = 0.0;
}
emxFree_real_T(&ry, 2);

// - 2 to eliminate first and last
i10 = ry->size[0] + ry->size[1];
ry->size[0] = iv6[0];
ry->size[1] = iv6[1];
emxEnsureCapacity((emxArray__common *)ry, i10, (int)sizeof(double));
414  k = iv6[0] * iv6[1];
415  for (i10 = 0; i10 < k; i10++) {
416    ry->data[i10] = 0.0;
417  }
418  
419  emxInit_real_T(&xAngs, 2);
420  i10 = xAngs->size[0] * xAngs->size[1];
421  xAngs->size[0] = iv6[0];
422  xAngs->size[1] = iv6[1];
423  emxEnsureCapacity((emxArray__common *)xAngs, i10, (int)sizeof(double));
424  k = iv6[0] * iv6[1];
425  for (i10 = 0; i10 < k; i10++) {
426    xAngs->data[i10] = 0.0;
427  }
428  
429  emxInit_real_T(&yAngs, 2);
430  i10 = yAngs->size[0] * yAngs->size[1];
431  yAngs->size[0] = iv6[0];
432  yAngs->size[1] = iv6[1];
433  emxEnsureCapacity((emxArray__common *)yAngs, i10, (int)sizeof(double));
434  k = iv6[0] * iv6[1];
435  for (i10 = 0; i10 < k; i10++) {
436    yAngs->data[i10] = 0.0;
437  }
438  
439  emxInit_real_T(&vecxP, 2);
440  i10 = vecxP->size[0] * vecxP->size[1];
441  vecxP->size[0] = iv6[0];
442  vecxP->size[1] = iv6[1];
443  emxEnsureCapacity((emxArray__common *)vecxP, i10, (int)sizeof(double));
444  k = iv6[0] * iv6[1];
445  for (i10 = 0; i10 < k; i10++) {
446    vecxP->data[i10] = 0.0;
447  }
448  
449  emxInit_real_T(&vecyP, 2);
450  i10 = vecyP->size[0] * vecyP->size[1];
451  vecyP->size[0] = iv6[0];
452  vecyP->size[1] = iv6[1];
453  emxEnsureCapacity((emxArray__common *)vecyP, i10, (int)sizeof(double));
454  k = iv6[0] * iv6[1];
455  for (i10 = 0; i10 < k; i10++) {
456    vecyP->data[i10] = 0.0;
457  }
458  
459  emxInit_real_T(&veczP, 2);
460  i10 = veczP->size[0] * veczP->size[1];
461  veczP->size[0] = iv6[0];
462  veczP->size[1] = iv6[1];
463  emxEnsureCapacity((emxArray__common *)veczP, i10, (int)sizeof(double));
464  k = iv6[0] * iv6[1];
465  for (i10 = 0; i10 < k; i10++) {
466    veczP->data[i10] = 0.0;
467  }
468  
469  // Final normal vector, all components
470  j = 1;
471  emxInit_real_T(&nArray, 2);
472  emxInit_real_T2(&c_yVec, 1);
473  emxInit_real_T2(&b_xVec, 1);
474  emxInit_real_T2(&c_xVec, 1);
475  emxInit_real_T2(&d_xVec, 1);
476  emxInit_real_T2(&e_xVec, 1);
477  emxInit_real_T2(&f_xVec, 1);
478  emxInit_real_T2(&g_xVec, 1);
479  emxInit_real_T2(&h_xVec, 1);
do {
  exitg3 = 0;
  i10 = c_yVec->size[0];
  c_yVec->size[0] = yVec->size[1];
  emxEnsureCapacity((emxArray__common *)c_yVec, i10, (int)sizeof(double));
  k = yVec->size[1];
  for (i10 = 0; i10 < k; i10++) {
    c_yVec->data[i10] = yVec->data[yVec->size[0] * i10];
  }
  i10 = c_yVec->size[0];
  if (j - 1 <= i10 - 3) {
    // Computes the cross-sectional angular normals to an input vector
    // normArray is an array of size [numel(vec), 2] correlating vector
    // from each positional index, 1)Magnitude & 2)Angle, computed from
    // from positive vector axis.
    i10 = b_xVec->size[0];
    b_xVec->size[0] = xVec->size[1];
    emxEnsureCapacity((emxArray__common *)b_xVec, i10, (int)sizeof(double));
    k = xVec->size[1];
    for (i10 = 0; i10 < k; i10++) {
      b_xVec->data[i10] = xVec->data[xVec->size[0] * i10];
    }
    i10 = b_xVec->size[0];
    n = nArray->size[0] * nArray->size[1];
    nArray->size[0] = i10 - 2;
    nArray->size[1] = 2;
    emxEnsureCapacity((emxArray__common *)nArray, n, (int)sizeof(double));
    k = (i10 - 2) << 1;
    for (i10 = 0; i10 < k; i10++) {
      nArray->data[i10] = 0.0;
    }
    // Eliminate Boundaries
    i10 = c_xVec->size[0];
    c_xVec->size[0] = xVec->size[1];
    emxEnsureCapacity((emxArray__common *)c_xVec, i10, (int)sizeof(double));
    k = xVec->size[1];
    for (i10 = 0; i10 < k; i10++) {
      c_xVec->data[i10] = xVec->data[xVec->size[0] * i10];
    }
    i10 = c_xVec->size[0];
    for (i = 0; i <= i10 - 3; i++) {
      n = d_xVec->size[0];
      d_xVec->size[0] = xVec->size[1];
      emxEnsureCapacity((emxArray__common *)d_xVec, n, (int)sizeof(double));
      k = xVec->size[1];
      for (n = 0; n < k; n++) {
        d_xVec->data[n] = xVec->data[xVec->size[0] * n];
      }
      n = e_xVec->size[0];
      e_xVec->size[0] = xVec->size[1];
      emxEnsureCapacity((emxArray__common *)e_xVec, n, (int)sizeof(double));
      k = xVec->size[1];
      for (n = 0; n < k; n++) {
        e_xVec->data[n] = xVec->data[xVec->size[0] * n];
      }
    }
    ang1 = atan(normCond->data[(i + normCond->size[0] * j) + 1] -
                normCond->data[i + normCond->size[0] * j]);
  }
}
// Angle from x-axis to (i-1)i
n = f_xVec->size[0];
f_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)f_xVec, n, (int)sizeof(double));
k = xVec->size[1];
for (n = 0; n < k; n++) {
    f_xVec->data[n] = xVec->data[xVec->size[0] * n];
}

n = g_xVec->size[0];
g_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)g_xVec, n, (int)sizeof(double));
k = xVec->size[1];
for (n = 0; n < k; n++) {
    g_xVec->data[n] = xVec->data[xVec->size[0] * n];
}

ang2 = atan(normCond->data[(i + normCond->size[0] * j) + 2] - normCond->data[(i + normCond->size[0] * j) + 1]);

// Angle from x-axis to i(i+1)
// normArray(i,1) = ang1; normArray(i,2) = ang2;
if ((ang1 >= 0.0) && (ang2 >= 0.0)) {
    ang = (3.1415926535897931 + ang1) - ang2) / 2.0 + ang2;
} else if ((ang1 >= 0.0) && (ang2 < 0.0)) {
    ang = (3.1415926535897931 + ang2) - ((3.1415926535897931 - ang1) + ang2) / 2.0;
} else if ((ang1 < 0.0) && (ang2 >= 0.0)) {
    ang = ((3.1415926535897931 - ang2) + ang1) / 2.0 + ang2;
} else {
    if ((ang1 < 0.0) && (ang2 < 0.0)) {
        ang = ((3.1415926535897931 - ang2) + ang1) / 2.0 + ang2;
    } else { // Correct
}

n = h_xVec->size[0];
h_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)h_xVec, n, (int)sizeof(double));
k = xVec->size[1];
for (n = 0; n < k; n++) {
    h_xVec->data[n] = xVec->data[xVec->size[0] * n];
}

n = i_xVec->size[0];
i_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)i_xVec, n, (int)sizeof(double));
k = xVec->size[1];
for (n = 0; n < k; n++) {
    i_xVec->data[n] = xVec->data[xVec->size[0] * n];
}

ndbl = normCond->data[i + normCond->size[0] * j] - normCond->data[(i + normCond->size[0] * j) + 1];
n = j_xVec->size[0];
j_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)j_xVec, n, (int)sizeof(double));
k = xVec->size[1];
for (n = 0; n < k; n++) {
    j_xVec->data[n] = xVec->data[xVec->size[0] * n];
}

n = k_xVec->size[0];
k_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)k_xVec, n, (int)sizeof(double));
k = xVec->size[1];
for (n = 0; n < k; n++) {
    k_xVec->data[n] = xVec->data[xVec->size[0] * n];
}

anew = normCond->data[(i + normCond->size[0] * j) + 1] - normCond->data[(i + normCond->size[0] * j) + 2];
newData = (sqrt(1.0 + ndbl * ndbl) + sqrt(1.0 + anew * anew)) / 2.0;
newData = (sqrt(1.0 + ndbl * ndbl) + sqrt(1.0 + anew * anew)) / 2.0;
}
k = nArray->size[0] - 1;
for (i10 = 0; i10 <= k; i10++) {
    ry->data[i10 + ry->size[0] * (j - 1)] = nArray->data[i10];
}
k = nArray->size[0] - 1;
for (i10 = 0; i10 <= k; i10++) {
    yAngs->data[i10 + yAngs->size[0] * (j - 1)] = nArray->data[i10 + nArray->size[0]];
}
j++;
} else {
    exitg3 = 1;
}
}
while (exitg3 == 0);

emxFree_real_T(&k_xVec);
emxFree_real_T(&j_xVec);
emxFree_real_T(&i_xVec);
emxFree_real_T(&h_xVec);
emxFree_real_T(&g_xVec);
emxFree_real_T(&f_xVec);
emxFree_real_T(&e_xVec);
emxFree_real_T(&d_xVec);
emxFree_real_T(&c_xVec);
emxFree_real_T(&b_xVec);
emxFree_real_T(&c_yVec);
i = 0;
emxInit_int32_T(&r1, 1);
emxInit_real_T(&b_normCond, 2);
emxInit_real_T2(&l_xVec, 1);
emxInit_real_T2(&l_yVec, 1);
emxInit_real_T2(&b_nArray, 1);
emxInit_real_T2(&c_nArray, 1);
do {
    exitg2 = 0;
    i10 = l_xVec->size[0];
l_xVec->size[0] = xVec->size[1];
emxEnsureCapacity((emxArray__common *)l_xVec, i10, (int)sizeof(double));
k = xVec->size[1];
for (i10 = 0; i10 < k; i10++) {
    l_xVec->data[i10] = xVec->data[xVec->size[0] * i10];
}
}
if (i <= i10 - 3) {
i10 = d_yVec->size[0];
d_yVec->size[0] = yVec->size[1];
emxEnsureCapacity((emxArray__common *)d_yVec, i10, (int)sizeof(double));
k = yVec->size[1];
for (i10 = 0; i10 < k; i10++) {
    d_yVec->data[i10] = yVec->data[yVec->size[0] * i10];
}

i10 = d_yVec->size[0];
b_normCond->size[0] = 1;
b_normCond->size[1] = k;
emxEnsureCapacity((emxArray__common *)b_normCond, i10, (int)sizeof(double));
for (i10 = 0; i10 < k; i10++) {
    b_normCond->data[b_normCond->size[0] * i10] = normCond->data[(i + normCond->size[0] * i10) + 1];
}
b_xSectionNormals(b_normCond, nArray);

k = rx->size[1];
i10 = r1->size[0];
r1->size[0] = k;
emxEnsureCapacity((emxArray__common *)r1, i10, (int)sizeof(int));
for (i10 = 0; i10 < k; i10++) {
    r1->data[i10] = i10;
}
k = nArray->size[0];
i10 = b_nArray->size[0];
b_nArray->size[0] = k;
emxEnsureCapacity((emxArray__common *)b_nArray, i10, (int)sizeof(double));
for (i10 = 0; i10 < k; i10++) {
    b_nArray->data[i10] = nArray->data[i10];
}

nm1d2 = r1->size[0];
for (i10 = 0; i10 < nm1d2; i10++) {
    rx->data[i + rx->size[0] * r1->data[i10]] = b_nArray->data[i10];
}
k = xAngs->size[1];
i10 = r1->size[0];
r1->size[0] = k;
emxEnsureCapacity((emxArray__common *)r1, i10, (int)sizeof(int));
for (i10 = 0; i10 < k; i10++) {
    r1->data[i10] = i10;
}
k = nArray->size[0];
i10 = c_nArray->size[0];
c_nArray->size[0] = k;
emxEnsureCapacity((emxArray__common *)c_nArray, i10, (int)sizeof(double));
for (i10 = 0; i10 < k; i10++) {
    c_nArray->data[i10] = nArray->data[i10 + nArray->size[0]];
}

nm1d2 = r1->size[0];
for (i10 = 0; i10 < nm1d2; i10++) {
    xAngs->data[i + xAngs->size[0] * r1->data[i10]] = c_nArray->data[i10];
}

++i;
else {
    exitg2 = 1;
}
while (exitg2 == 0);
i0 = 0;
emxFree_real_T(&m_xVec);
emxFree_real_T(&yAngs);
emxFree_real_T(&xAngs);
emxFree_real_T(&ry);
emxFree_real_T(&rx);
emxFree_real_T(&xVec);
// If non-integer position
VqX = interp2(vecxP, 1.0 + (v3[0] - floor(v3[0])), 1.0 + (v3[1] - floor(v3[1])));

// Find weighted normal, x
VqY = interp2(vecyP, 1.0 + (v3[0] - floor(v3[0])), 1.0 + (v3[1] - floor(v3[1])));
// Find weighted normal, y
VqZ = interp2(veczP, 1.0 + (v3[0] - floor(v3[0])), 1.0 + (v3[1] - floor(v3[1])));

// Find weighted normal, z
normalVec[0] = VqX;
normalVec[1] = VqY;
normalVec[2] = VqZ;
anew = norm(normalVec);
emxFree_real_T(&veczP);
emxFree_real_T(&vecyP);
emxFree_real_T(&vecxP);
for (i9 = 0; i9 < 3; i9++) {
    normalVec[i9] /= anew;
}
}
APPENDIX 2 – SUBMITTED PAPERS

The following primary-authored papers provided are as follows:


Additional co-authored publications not provided in this appendix:


Bibliography


CURRICULUM VITÆ

Dale E. Karas Ph.D.

Curriculum Vitae

U.S. Citizen – Ph.D. Graduation Date: 12 May 2018

Profile: CV usage for dissertation submission, Thu, 12 April 2018, in part fulfilling requirements for the Doctor of Philosophy: Mechanical Engineering degree.

Education

Dissertation Title: Morphological & Energy Transport Optimization of Spectrally-Selective Solar Absorber Coatings at Mesoscale
+Graduate College Research & Mentorship Certifications

+Chemical Engineering Minor, Mathematics Minor

+Piano/Organ Performance Emphasis

Honors & Awards

2018 20th Annual Graduate & Professional Student Research Forum - 1st Place Poster
2014,2017 OSA: The Optical Society - Congressional Visits Day (Travel Grant Recipient 2014, promoted support for photonics technologies and R&D to members of Congress)
2014 SPIE: The International Society for Optics and Photonics - 2014 Optics Outreach Games, People’s Choice Award (Rep. UA-OSC Student Optics Chapter)
2012–2013 Jack D. Gaskill Endowed Scholarship
2011–2013 John Tipton Scholarship in Optical Sciences
2009–2011 The University of Arizona - Academic Distinction, Dean’s List — BS/BM
Professional Experience

Vocational

2015–Present 
**Graduate Student Researcher & Ph.D. Candidate**, *The Center for Energy Research at The University of Nevada, Las Vegas (UNLV)*, Las Vegas, NV.
Supported energy-efficient materials design, fabrication, and analysis of thermoelectric materials, coordinated design and construction of prototype solar distillation and CSP systems, commenced laboratory study and transition into infrared coating materials research and additive manufacturing research.

2014 
**Graduate Student Researcher**, *Large Optics Fabrication & Testing (LOFT) Group, Steward Observatory Mirror Lab*, Tucson, AZ.
Developed processing and analysis routines for computing aspheric optical field positions, IR-PSF data, and Fourier domain representation of sequential raytrace and illumination design for Large Synoptic Survey Telescope (LSST) mirror components.

2014 
**Graduate Research Intern/Member of Technical Staff**, *Integrated Sensor Design & Analysis Department, The Aerospace Corporation*, El Segundo, CA.
Characterized focal plane arrays, refined computational models for IR hyperspectral sensor design simulation and analysis, and administered state-of-the-art metrological optical testing for remote sensing technologies.

2013–14 
**Graduate Student Researcher**, *Advanced Sensing Lab (ASL)*, *University of Arizona College of Optical Sciences*, Tucson, AZ.
Developed illumination scheme, design, and optimization metrics for Sunkist’s Research & Technical Division fruit graters along with phenomenological product classification and sorting algorithmic elaboration for machine-learning computer vision studies.

2012 
**Undergraduate Student Researcher**, *Center for Energy Efficient Electronics Science (E3S) & Center for Integrated Access Networks (CIAN) Optoelectronics Research Experience for Undergraduates (REU)*, *The University of California, Berkeley*, Berkeley, CA.
Fabricated and tested silicon-photonic waveguides as hybridized in a LIDAR system with VCSEL and CMOS components for detection of varying optical communication systems’ runtime measurements, spearheaded analytical modeling and test for project design interferometric and polarimetric signal phenomenology in various electro-optical design configurations.

2011 
**Undergraduate Student Researcher**, *Nanophotonic Materials Group*, *CREOL: The College of Optics & Photonics, The University of Central Florida*, Orlando, FL.
Fabricated nanophotonic materials, semiconductor media, and photonics crystals. Gained involvement with academic directed research in chemical engineering and optical sciences; developed expertise with photolithography, SEM, AA, statistical sampling, and spectrophotometric experimentation.

2010 
**Intern**, *Tronox, LLC.*, Henderson, NV.
Supported engineering field research and chemical laboratory research for production process and maintenance yield of patented chemicals at Tronox LLC Henderson chemical plant. Expanded record tracking resources by authoring algorithms/macros to streamline digitization of archived databases.
Miscellaneous

2013–15 **Assistant Music Director,** *Act 1 Academy of Performing Arts,* Tucson, AZ.
Directed K-12 level students in music performance and musical theater productions, and provided instruction for music performance, technology, and composition. Prepared music transcriptions and personal compositions for performances, coordinated rehearsals and stage productions.

2012–14 **Guest Composer,** *Information Technologies Specialist, Arizona Friends of Chamber Music & Reveille Men’s Chorus,* Tucson, AZ.
Supported the Arizona Friends of Chamber Music and Reveille Men’s Chorus board of directors as a technological specialist, music arranger, and programmer. Coordinated strategies for audience development, marketing, advertiser & sponsor outreach, coordination, and communication.

2011–12 **Student Curator,** *The Museum of Optics, Optical Sciences Center,* Tucson, AZ.
Employed as a museum curator and information systems specialist for the UA College of Optical Science’s established Museum of Optics. Provided web design and photography services, and developed a newer SQL database infrastructure to overhaul collection items for port.

2009-13 **Music Instructor,** *[Various],* Tucson, AZ & Las Vegas, NV.
Worked as a private music instructor for several students who coordinated through University or online services. Collaborated with numerous music chamber ensembles as a pianist and vocalist, including classical, jazz, dance, and musical theatre repertoire.

Professional Associations

2017–Present IEEE, Institute of Electrical and Electronics Engineers
2017–Present MRS, The Materials Research Society
2016–Present ANS, The American Nuclear Society
2015–Present ASME, The American Society of Mechanical Engineers
2013–Present ADN, The Autodesk Developer Network
2010–Present SPIE, The International Society of Optics & Photonics
2010–Present OSA, The Optical Society
2009–Present ACS, The American Chemical Society

Competencies & Qualifications

**Programming Languages + Developer Tools**

- **Basic** Wordpress & Drupal CMS, LAMP Stack (Linux, Apache, MySQL, PHP)
- **Intermediate** c/c++/c#, LabVIEW, R, BASIC, PYTHON, HTML/CSS
- **Advanced** MATLAB, GNU Components: Octave & GNUPlot, \LaTeX, UNIX, Computer Hardware + Information Technologies Support

4505 S. Maryland Pkwy, Box. 45-4027 – Las Vegas, NV 89154-4027
📞 (702) 527 4583 • 📞 (702) 895 3936 • ✉️ dalekaras@gmail.com
✉️ eeml.egr.unlv.edu • The University of Nevada, Las Vegas
General Design/Engineering Software

**Autodesk Design Suite**: Inventor, Simulation Multiphysics + Computational Fluid Dynamics, AutoCAD, Fusion; **Dassault Systèmes**: SolidWorks, CATIA; **Mathworks**: MATLAB, Simulink; **Wolfram Research**: Mathematica; **Zemax**: OpticStudio, ZEMAX; **Photon Engineering**: FRED; **Breault Research Organization (BRO)**: ASAP, APEX; **Adobe**: Photoshop, Illustrator, InDesign, Fireworks, Premier, Audition, Dreamweaver; **Microsoft**: Word, Excel (w/VBA), PowerPoint, Access, OneNote, Project, Visio, Visual Studio

Instrumentation Experience

**Mechanical**: Lathe, Mill, Drill Press, Grinders, Metrological Instrumentation, CAD/CAM, CNC Programming; **Chemical Laboratory**: OS/AES/MS/MFS/AAS, GC/LC/IC, Colorimetry, Voltammetry, Electrochemical Titration + Bench Cell Sensor Design, Calorimetry, Chemical Fractionation, Direct Laser Writing; **Optical**: Microscopy (SEM/TEM/Fluorescence), Infrared (FPAs, Photodetectors, Bolometers), Interferometry (Fizeau, Mach-Zehnder, Twyman-Green, PSI), Laser Systems (HeNe, CO2, Ti:Sapphire, Nd:YAG, VCSELs), Polarimetry, Hyperspectral Imaging, E-Beam Photolithography, Spectroscopic Systems (Raman, Scanning Monochromator), Scintillation

Certifications

2011,’13,’15 **Photon Engineering**: FRED Optical Engineering Software Tutorial, Principles of Stray Light Short Course, Physical Optics Modeling Short Course
2014 **Breault Research Organization (BRO)**: ASAP Introductory Tutorial + Advanced Illumination Design Tutorial
2012 **UA College of Optical Sciences**: Optics of Imaging in Lithography Workshop + Optical Fabrication & Testing Workshop Certifications
2009-17 **[Various]**: Chemical Hygiene/Laboratory Safety + PPE Handling, Radiation Safety, ESD, Cryogenics Handling, Laser Safety, Fire Prevention & Control, AED+CPR, First Aid, Hazard Communication, Hazardous Waste Handling

Publications

2018 H. An, D. Karas, B. Kim, S. Trabia, J. Moon, “*Flexible n-type thermoelectric composite films with enhanced performance through interface engineering and post-treatment*,” IOP Nanotechnology, [Accepted].

Oral Presentations


2017 Spectrally-Selective Copper-Oxide Spinel Absorber Coatings for High-Temperature Concentrated Solar Power Systems – UNLV Graduate & Professional Student Association (GPSA) Research Forum; Las Vegas, NV

Poster Presentations

2018 UNLV Graduate & Professional Student Association (GPSA) Research Forum; Las Vegas, NV

2016,2017 ACS Poster Contest & Exposition, Southern Nevada Section; Henderson, NV

2017 Material Synthesis & Optical Simulation of High-Temperature Solar Absorber Coating Materials – 25th National Science Foundation (NSF) Established Program to Stimulate Competitive Research (EPSCoR) Conference; Missoula, MT

2017 Solar-Energy-Water-Environment NEXUS in Nevada Annual Meeting; Las Vegas, NV

2016 Solar-Energy-Water-Environment NEXUS in Nevada Annual Meeting; Reno, NV

2013 UA-OSC SPIE/OSA Student Optics Chapter – The International Society of Optics & Photonics (SPIE), Optics+Photonics 2014 Conference; San Diego, CA


2009 University of Arizona SAACS (Student Affiliates of the American Chemical Society): Stronger, Better, Faster, Greener – 237th American Chemical Society (ACS) National Meeting; Salt Lake City, UT
Additional Conferences & Outreach

2015, 2017  Autodesk University; Las Vegas, NV
2017  Clean Energy Summit; Las Vegas, NV
2017  Solar Power International; Las Vegas, NV
2017  Solar Power International; Las Vegas, NV
2013, 2014  OSA Frontiers in Optics/Laser Science; Orlando, FL & Tucson, AZ
2013, 2014  SPIE Optics + Photonics; San Diego, CA

Languages

Primary  **English, C++**  Complete read/write/execute fluency
Intermediate  **Greek, French**  Conversationally fluent
Basic  **Spanish, Chinese 中文 (Mandarin 普通话)**  Basic words and phrases only