RADIATIVE TRANSPORT IN PLANT CANOPIES: FORWARD AND INVERSE PROBLEM FOR UAV APPLICATIONS

By

Roberto Furfaro

A Dissertation Submitted to the Faculty of the
AEROSPACE AND MECHANICAL ENGINEERING DEPARTMENT
In Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN AEROSPACE ENGINEERING

In the Graduate College
THE UNIVERSITY OF ARIZONA

2004
As members of the Final Examination Committee, we certify that we have read the
dissertation prepared by Roberto Furfaro

entitled Radiative Transport in Plant Canopies: Forward and Inverse Problem for UAV

Applications

and recommend that it be accepted as fulfilling the dissertation requirement for the

Degree of Doctor of Philosophy

Barry Ganapol

John Williams

Ara Arabyan

Final approval and acceptance of this dissertation is contingent upon the candidate’s
submission of the final copy of the dissertation to the Graduate College.

I hereby certify that I have read the dissertation prepared under my direction and
recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director: Barry Ganapol
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under the rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Roberto Turbino
ACKNOWLEDGEMENTS

This dissertation is the completion of what was a fantastic and intense three-year journey created and designed by Professor Barry Ganapol. He taught me the fine and precious art of researching and therefore he is the first person I would like to acknowledge: He introduced me to the scientific realm of transport theory and encouraged me to expand my knowledge and understanding as well as he pushed me toward new and exciting challenges.

I would like to acknowledge Lee Johnson who provided guidance and support during my time spent at NASA Ames. His remarks and observations were extremely helpful to clarify some important issues arising during the ground and airborne image analysis.

Moreover, I would like to acknowledge Dr. Stan Herwitz, director of the NASA UAV Coffee Project who welcomed me and supported me as part of the imaging team.

Last but definitely not the least, I would like to acknowledge my parents, my sister and my wonderful “soon-to-be wife” Elise for the encouragement, the understanding and the moral support always provided. While going through the various stages of this research I realized how crucial is to have a supporting family to share dreams and overcome difficulties. My success in completing this dissertation is also their success.
DEDICATION

This dissertation is dedicated to Elise, Graziella and my parents Rocco and Bettina.
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF FIGURES</td>
<td>8</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>12</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>15</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>17</td>
</tr>
<tr>
<td>1.1 Radiative transport and remote sensing</td>
<td>17</td>
</tr>
<tr>
<td>1.2 Radiative transfer model in dense plant canopies and Neural Networks: Inverse problem and applications</td>
<td>19</td>
</tr>
<tr>
<td>1.3 NASA “Coffee Project”: Searching for heavenly coffee</td>
<td>21</td>
</tr>
<tr>
<td>1.4 Outline of the research to be presented</td>
<td>26</td>
</tr>
<tr>
<td>2. PART I: MODELING THE RADIATIVE REGIME IN DENSE CANOPY STRUCTURES</td>
<td>28</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>28</td>
</tr>
<tr>
<td>2.2 Level One: Modeling the radiative regime within leaf (LEAFMOD derivates)</td>
<td>29</td>
</tr>
<tr>
<td>2.2.1 LEAFMOD model description: One-angle radiative transfer equation</td>
<td>33</td>
</tr>
<tr>
<td>2.2.2 LEAFMOD two-angle model: Governing equation</td>
<td>35</td>
</tr>
<tr>
<td>2.3 Level Two: Modeling the radiative regime within the canopy (Two-angle CANMOD)</td>
<td>36</td>
</tr>
<tr>
<td>2.3.1 CANMOD two-angle formulation: Model description</td>
<td>38</td>
</tr>
<tr>
<td>2.3.2 CANMOD one-angle formulation: Summary</td>
<td>44</td>
</tr>
<tr>
<td>2.4 Level Three: Coupling Leaf and Canopy model (LCM2)</td>
<td>47</td>
</tr>
<tr>
<td>3. PART II: SPECTRAL THEORY FOR THE CANOPY EQUATION</td>
<td>51</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>51</td>
</tr>
<tr>
<td>3.2 Spectral theory for canopies with separable kernel: Formulation</td>
<td>53</td>
</tr>
<tr>
<td>3.3 Singular Eigenfunctions Method: Caseology for the canopy transport equation</td>
<td>57</td>
</tr>
<tr>
<td>3.3.1 Caseology: Separation of variables and the eigenvalue problem</td>
<td>58</td>
</tr>
<tr>
<td>3.3.2 Caseology: Discrete spectrum</td>
<td>60</td>
</tr>
<tr>
<td>3.3.3 Dispersion relationship for the canopy transport equation</td>
<td>62</td>
</tr>
<tr>
<td>3.3.4 Caseology: Continuous spectrum</td>
<td>65</td>
</tr>
<tr>
<td>3.3.5 Boundary values of the dispersion function</td>
<td>69</td>
</tr>
<tr>
<td>3.3.6 Orthogonality of the discrete and continuous eigenfunctions</td>
<td>71</td>
</tr>
<tr>
<td>3.3.7 Completeness of the complete set of eigenfunctions</td>
<td>72</td>
</tr>
<tr>
<td>3.4 Spectral Theory and FN method</td>
<td>78</td>
</tr>
<tr>
<td>4. PART III: NUMERICAL SOLUTION OF THE LEAF AND CANOPY EQUATIONS. DISCRETE ORDINATE METHOD</td>
<td>82</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>82</td>
</tr>
<tr>
<td>4.2 Numerical solution for LEAFMOD: Discrete Ordinate Method for one-angle formulation</td>
<td>84</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS - Continued

4.2.1. Numerical implementation via zero weights concept: Comparison with the FN solution .......................................................... 90
4.3. Discrete Ordinate Method for CANMOD: One-angle formulation ................. 97
  4.3.1. Numerical implementation: Comparison CANMOD SN versus FN solution 103
4.4. Discrete Ordinate Method for CANMOD: Two-angle formulation ........... 111
  4.4.1. Numerical implementation: Two-angles versus One-angle ................. 119

5. PART IV: LEAF-CANOPY MODEL BASED NEURAL NETWORK SOLUTION TO THE INVERSE PROBLEM .................................................. 128
  5.1. Introduction ............................................................................. 128
  5.2. Multilayer Neural network using back-propagation: Theory ..................... 133
  5.3. Independent retrieval of canopy variables: Methods and results ............ 136
    5.3.1. LAI independent retrieval: design, training and testing results ....... 138
    5.3.2. Independent chlorophyll retrieval: Design, training and testing results 146
  5.4. Neural Network inversion: Sensitivity analysis ................................ 153

6. PART V: UAV "COFFEE PROJECT": MODEL-BASED NEURAL NETWORKS FOR COFFEE RIPENESS ESTIMATION ................................................. 163
  6.1. Introduction ............................................................................. 163
  6.2. LCM2 extension: Coffee cherry spectra ........................................... 164
  6.3. LCM2 inversion via neural network for ripeness estimation: Design and Training ............................................................................ 167
  6.4. Ripeness estimation via neural network: Network performance and stability ... 177
  6.5. Neural networks for ripeness predictions: Ground images .................... 180
    6.5.1. DN-Reflectance transformation: Problems and solutions ............. 182
    6.5.2. Pure cherry study and sensitivity analysis ................................ 192
    6.5.3. Building the interface for Reflectance-DNs: a new strategy ......... 195
    6.5.4. Neural Network Test: Pure cherry predictive capability ............. 197
    6.5.5. Ripeness Estimation for the side-loaded image ......................... 201
  6.6. Neural networks for ripeness prediction: Airborne images .................. 210
    6.6.1. Domain Projection Technique (DPT): Outline of the method ........ 213
    6.6.2. Domain Projection Technique (DPT): Testing the method ............ 220
    6.6.3. Neural Network ripeness estimation: DPT application and results ... 223

7. SUMMARY AND CONCLUSION ................................................................ 238

APPENDIX A ...................................................................................... 245
REFERENCES .................................................................................... 257
LIST OF FIGURES

Figure 1: Helios and Pathfinder Plus flying in the Hawaiian sky ........................................ 24
Figure 2: Conceptual scheme for the on-board network operations. Images are taken by the camera system and processed by the neural network. The results is sent to the ground via telemetry .................................................. 26
Figure 3: From modeling to application: Solution scheme for the “End-to-End” problem .... 27
Figure 4: Leaf idealized as a slab of optical material infinite in transverse direction ....... 33
Figure 5: Leaf interior scattering process. Complex light-cells interactions tend to average .................................................. 33
Figure 6: Canopy as “turbid” medium ........................................................................ 37
Figure 7: LAD as function of the leaf normal inclination angle .............................................. 41
Figure 8: Idealization of the leaf optical behavior. bi-Lambertian model ......................... 42
Figure 9: LCM2 flow chart. Connection between LEAFMOD and CANMOD ............ 50
Figure 10: Leaf spatial discretization scheme ................................................................. 87
Figure 11: LEAFMOD FN and SN computed Reflectance at various wavelengths and spatial discretization effect ................................................................. 93
Figure 12: LEAFMOD FN and SN computed Transmittance at various wavelengths and spatial discretization effect ................................................................. 93
Figure 13: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of spatial points as parameter) .................................................. 94
Figure 14: Comparing FN and SN results. Transmittance absolute error at various wavelengths (number of spatial points as parameter) .................................................. 94
Figure 15: LEAFMOD FN and SN computed Reflectance at various wavelengths and angular discretization effect ................................................................. 95
Figure 16: LEAFMOD FN and SN computed Transmittance at various wavelengths and angular discretization effect ................................................................. 95
Figure 17: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of angular rays as parameter) .................................................. 96
Figure 18: Comparing FN and SN results. Transmittance absolute error at various wavelengths (number of angular rays as parameter) .................................................. 96
Figure 19: $H(\mu, \mu_c)$ contour plot .............................................................................. 99
Figure 20: Canopy spatial discretization scheme ............................................................... 101
Figure 21: CANMOD FN and SN computed Reflectance at various wavelengths and angular discretization effect ................................................................. 106
Figure 22: CANMOD FN and SN computed Reflectance at various wavelengths and spatial discretization effect ................................................................. 106
Figure 23: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of angular rays as parameter) .................................................. 107
Figure 24: Comparing FN and SN results. Reflectance relative error at various wavelengths (number of angular rays as parameter) .................................................. 107
Figure 25: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of spatial points as parameter) .................................................. 108
LIST OF FIGURES - Continued

Figure 26: Comparing FN and SN results. Reflectance relative error at various wavelengths (number of spatial points as parameter) ........................................... 108
Figure 27: Unit sphere. The photon direction is described by azimuth and inclination. 115
Figure 28: Area scattering function contour plot .................................................. 115
Figure 29: Sweeping technique sequence of operations ....................................... 118
Figure 30: Radiant intensity polar plot for the sun at the zenith and planophile canopy architecture. The various lines correspond to different inclination angles .......... 123
Figure 31: Radiant intensity polar plot for the extremophile canopy architecture. The various lines correspond to different inclination angles .................. 126
Figure 32: Radiant intensity polar plot for the 13 degree inclination angle output and extremophile case ................................................................. 126
Figure 33: Radiant intensity polar plot for all possible inclination angles at the top-of-the-canopy in the extremophile case ............................................. 127
Figure 34: Example of multilayer neural network ............................................... 136
Figure 35: Reflectance as function of LAI with wavelength as parameter. The plot was generated using LCM2 .................................................. 141
Figure 36: Training session results. MSE as function of number of epochs for the six networks trained on larger and larger training sets .............................. 142
Figure 37: Neural network post-processing analysis. Linear fit ....................... 143
Figure 38: Training session results for the chlorophyll retrieval design. MSE as function of number of epochs for the six networks trained on larger and larger training sets .................................................. 148
Figure 39: Post-processing analysis for the chlorophyll case. Linear fit .................. 149
Figure 40: Training session on corrupted data for LAI retrieval ............................ 156
Figure 41: Post-regression analysis and linear fit for LAI retrieving neural network trained on corrupted data ................................................................. 157
Figure 42: Comparison between LAI true value (red circles) and LAI retrieved values (blue line). Corruption noise level was 2.5% ............................................. 159
Figure 43: Comparison between LAI true value (red circles) and LAI retrieved values (blue line). Corruption noise level was 5% ............................................. 160
Figure 44: Comparison between chlorophyll true value (red circles) and chlorophyll retrieved values (blue line). Corruption noise level was 2.5% .................. 160
Figure 45: Comparison between chlorophyll true value (red circles) and chlorophyll retrieved values (blue line). Corruption noise level was 2.5% .................. 161
Figure 46: Coffee Cherries measured spectral reflectance .................................. 166
Figure 47: Neural network “black box” scheme ............................................... 170
Figure 48: Class-one neural network training session. MSE as function of number of epochs. ....................................................................................... 174
Figure 49: Post-processing analysis. Linear fit for under-ripe percentage network retrieval ................................................................. 174
Figure 50: Post-processing analysis. Linear fit for ripe percentage network retrieval. .. 175
LIST OF FIGURES - Continued

Figure 51: Post-processing analysis. Linear fit for over-ripe percentage network retrieval. ................................................................. 175
Figure 52: Class-two neural network training session. MSE as function of number of epochs. ......................................................... 176
Figure 53: Post-processing analysis. Linear fit for leaf and under-ripe percentage network retrieval ........................................... 176
Figure 54: Post-processing analysis. Linear fit for ripe and over-ripe percentage network retrieval .............................................. 177
Figure 55: False color image of the reflectance computed via LCM2 of the coffee-canopy simulated scene ........................................ 178
Figure 56: Side-loaded truck image ............................................................................................................................................. 183
Figure 57: Transformed reflectance (red points) falling outside the training set (blue points) ............................................................. 184
Figure 58: DN-reflectance map. Cubic interpolation at the three band of interest ............................................................................. 187
Figure 59: Averaged truck image after considering only transformed pixels .......................................................... 188
Figure 60: Averaged truck image. Zoom on the top-left 5x5 area ................................................................................................. 189
Figure 61: Averaged 100x100 image after DN-Reflectance transformation .................................................................................. 191
Figure 62: Sampled yellow cherry .................................................................................................................................................. 192
Figure 63: DN-Reflectance transformation for the three visible bands .......................................................................................... 197
Figure 64: Green (under-ripe) cherry sampled from the truck image ............................................................................................ 199
Figure 65: Red (Over-ripe) cherry sampled from the truck image ................................................................................................. 199
Figure 66: 10x10 averaged imaged of the transformed pixels ........................................................................................................ 202
Figure 67: Mostly ripe sub-image ................................................................................................................................................... 204
Figure 68: Mostly under-ripe sub-image .................................................................................................................................. 204
Figure 69: Mostly over-ripe sub-image ........................................................................................................................................ 205
Figure 70: Side-loaded truck image #2 ....................................................................................................................................... 205
Figure 71: 10x10 averaged and transformed truck image #2 ...................................................................................................... 207
Figure 72: Side-loaded truck. Sub-image #1 ................................................................................................................................... 208
Figure 73: Side-loaded truck. Sub-image #2 ................................................................................................................................... 208
Figure 74: Side-loaded truck. Sub-image #3 ................................................................................................................................... 209
Figure 75: Collage of images taken by the Helios UAV camera system. The collage partially reconstruct the Kauai coffee field ........................................................................................................................................... 211
Figure 76: Training set in the reflectance space .............................................................................................................................. 215
Figure 77: Domain projection demonstration .............................................................................................................................. 218
Figure 78: 4x4 simulated coffee canopy scene ................................................................................................................. 221
Figure 79: Position of the 406, 408 and 410 fields ......................................................................................................................... 228
Figure 80: Field 408 ........................................................................................................................................................................ 228
Figure 81: Field 406 ........................................................................................................................................................................ 229
Figure 82: Field 410 ........................................................................................................................................................................ 229
Figure 83: Field 408 Block 1 ......................................................................................................................................................... 233
Figure 84: Field 408 Block 2 ......................................................................................................................................................... 234
Figure 85: Field 408 Block #3 ................................................................. 245
Figure 86: Field 408 Block #4 ................................................................. 246
Figure 87: Field 406 Block #1 ................................................................. 246
Figure 88: Field 406 Block #2 ................................................................. 247
Figure 89: Field 406 Block #3 ................................................................. 247
Figure 90: Field 406 Block #4 ................................................................. 248
Figure 91: Field 406 Block #5 ................................................................. 248
Figure 92: Field 406 Block #6 ................................................................. 249
Figure 93: Field 406 Block #7 ................................................................. 249
Figure 94: Field 406 Block #8 ................................................................. 250
Figure 95: Field 406 Block #9 ................................................................. 250
Figure 96: Field 406 Block #10 ............................................................... 251
Figure 97: Field 406 Block #11 ............................................................... 251
Figure 98: Field 406 Block #12 ............................................................... 252
Figure 99: Field 406 Block #13 ............................................................... 252
Figure 100: Field 406 Block #14 ............................................................. 253
Figure 101: Field 410 Block #1 ............................................................... 253
Figure 102: Field 410 Block #2 ............................................................... 254
Figure 103: Field 410 Block #3 ............................................................... 254
Figure 104: Field 410 Block #4 ............................................................... 255
Figure 105: Field 410 Block #5 ............................................................... 255
Figure 106: Field 410 Block #6 ............................................................... 256
## LIST OF TABLES

Table 1: Leaf Angle Distribution Function (LAD). Common canopy architectures and related $g_L(\mu_t) = g_L(\phi)$ functions ................................................................. 40

Table 2: CANMOD input data for maple-canopy simulation scenes ........................................... 104

Table 3: Comparing FN and SN. Results and absolute errors for various angular discretization points ................................................................. 109

Table 4: Comparing FN and SN. Results and absolute errors for various spatial discretization points ................................................................................................. 109

Table 5: MATLAB SN implementation: CPU time ...................................................................... 110

Table 6: Canopy input data for one-angle and two angle CANMOD simulation.......... 120

Table 7: Hemispherical reflectance computation via one-angle implementation ........ 121

Table 8: Hemispherical reflectance computation via two-angle implementation ...... 121

Table 9: Comparison of one-angle and two angle results. Error analysis ..................... 121

Table 10: Effect of the azimuth on hemispherical reflectance computation ............. 121

Table 11: Radiant intensity at the top-of-the-canopy as function of azimuth and inclination (Planophile case) ................................................................. 122

Table 12: Radiant intensity at the top-of-the-canopy as function of azimuth and inclination (Extremophile case) ................................................................. 125

Table 13: Training results summary ....................................................................................... 144

Table 14: Canopy scene simulation and LAI retrieval results ........................................... 145

Table 15: LAI network retrieval error for the LCM2 simulated canopies ............... 145

Table 16: Relative error percentage for LAI retrieval ......................................................... 146

Table 17: Root Mean Square Error comparison ................................................................. 146

Table 18: Summary of training results for the chlorophyll case ........................................ 147

Table 19: Canopy scene simulation and chlorophyll retrieval results ....................... 150

Table 20: Chlorophyll network retrieval error for the LCM2 simulated canopies .... 151

Table 21: Relative error percentage for chlorophyll retrieval .................................. 152

Table 22: RMSE comparison for chlorophyll retrieval .................................................. 152

Table 23: 18 canopy scenes for neural network, trained on corrupted data, performance analysis ....................................................................................... 158

Table 24: RMSE comparison with data available from the literature ......................... 162

Table 25: Input data for the coffee-canopy simulated scene ........................................... 178

Table 26: Real and predicted leaf and coffee-types percentage for the 4x4 canopy scene. ....................................................................................... 179

Table 27: Absolute neural network retrieval error .......................................................... 180

Table 28: LCM2 computed reflectance for pure cherries ......................................................... 185

Table 29: Average DNs for pure cherries extracted from the image ............................... 185

Table 30: DNs of the pure cherries are associated to the correspondent reflectance in the red band ....................................................................................... 185

Table 31: DNs of the pure cherries are associated to the correspondent reflectance in the green band ....................................................................................... 186
LIST OF TABLES - Continued

Table 32: DNs of the pure cherries are associated to the correspondent reflectance in the blue band. ................................................................. 186
Table 33: Neural over-ripe percentage estimation of the top-left 5x5 area .................. 189
Table 34: Neural ripe percentage estimation of the top-left 5x5 area ....................... 189
Table 35: Neural under-ripe percentage estimation of the top-left 5x5 area ............... 190
Table 36: Neural network percentage estimation for the sampled yellow cherry ........ 193
Table 37: Sampled yellow cherry and baseline yellow cherry used in the transformation. ................................................................. 193
Table 38: Network response under reflectance increasing perturbation ..................... 194
Table 39: Network response under reflectance decreasing perturbation .................... 194
Table 40: Reflectance-DN at the red band ...................................................... 196
Table 41: Reflectance-DN at the green band .................................................... 196
Table 42: Reflectance-DN at the blue band ..................................................... 196
Table 43: Network prediction over pure sampled cherries ..................................... 198
Table 44: Neural network cherry percentages estimation .................................... 202
Table 45: Comparison between hand count and neural network results (mostly ripe)... 204
Table 46: Comparison between hand count and neural network results (mostly under-ripe). ........................................................................ 204
Table 47: Comparison between hand count and neural network results (mostly over-ripe). ........................................................................ 205
Table 48: Neural network cherries estimation. Tuning and no-tuning effect ............. 207
Table 49: Comparison between neural network prediction and hand count (sub-image #1) ........................................................................ 208
Table 50: Comparison between neural network prediction and hand count (sub-image #2). ........................................................................ 208
Table 51: Comparison between neural network prediction and hand count (sub-image #3). ........................................................................ 209
Table 52: Domain projection demonstration. The system is driven from unphysical to physical ................................................................. 218
Table 53: Real and computed percentages for the 4x4 simulated scene .................... 222
Table 54: Global network error ........................................................................ 222
Table 55: Table of the a-priori input parameters for generating the 27 class-two neural networks ........................................................................ 225
Table 56: Parchment data (Kauai Coffee Company) ............................................. 227
Table 57: Block neural network percentage estimation results for the coffee field 408. 232
Table 58: Block neural network percentage estimation results for the coffee field 406. ........................................................................ 232
Table 59: Block neural network percentage estimation results for the coffee field 410. 233
Table 60: Comparison between neural network prediction, parchment data and branch count for field 408. ................................................................. 235
Table 61: Comparison between neural network prediction, parchment data and branch count for field 406................................................................. 236
Table 62: Comparison between neural network prediction, parchment data and branch count for field 410................................................................. 236
Table 63: Neural networks used in percentage prediction for field 408....................... 237
Table 64: Neural networks used in percentage prediction for field 406....................... 237
Table 65: Neural networks used in percentage prediction for field 410....................... 237
ABSTRACT

This dissertation deals with modeling the radiative regime in vegetation canopies and the possible remote sensing applications derived by solving the forward and inverse canopy transport equation. The aim of the research is to develop a methodology (called "end-to-end problem solution") that, starting from first principles describing the interaction between light and vegetation, constructs, as the final product, a tool that analyzes remote sensing data for precision agriculture (ripeness prediction). The procedure begins by defining the equations that describe the transport of photons inside the leaf and within the canopy. The resulting integro-differential equations are numerically integrated by adapting the conventional discrete-ordinate methods to compute the reflectance at the top of the canopy. The canopy transport equation is also analyzed to explore its spectral properties. The goal here is to apply Case's method to determine eigenvalues and eigenfunctions and to prove completeness.

A model inversion is attempted by using neural network algorithms. Using input-outputs generated by running the forward model, a neural network is trained to learn the inverse map. The model-based neural network represents the end product of the overall procedure.

During Oct 2002, an Unmanned Aerial Vehicles (UAVs) equipped with a camera system, flew over Kauai to take images of coffee field plantations. Our goal is to predict the amount of ripe coffee cherries for optimal harvesting. The Leaf-Canopy model was modified to include cherries as absorbing and scattering elements and two classes of neural networks were trained on the model to learn the relationship between reflectance
and percentage of ripe, over-ripe and under-ripe cherries. The neural networks are interfaced with images coming from Kauai to predict ripeness percentage. Both ground and airborne images are considered. The latter were taken from the on-board Helios UAV camera system flying over the Kauai coffee field. The results are compared against hand counts and parchment data to evaluate the network performances on real applications. In ground images, the error is always less than 11%. In airborne image, the error bound is 20%.

The results are certainly adequate and show the tremendous potential of the methodology.
1. INTRODUCTION

1.1. Radiative transport and remote sensing

Understanding how light interacts with vegetation is a problem of paramount importance in several scientific communities. For example, the past decade witnessed the increased need for quantifying the radiant energy distribution in vegetation canopy as a mean for disclosing and understanding the underlying biogeochemical processes driving the interaction between various ecosystems. While satellite remote sensing data are largely made available to the scientific world throughout a web-based distribution, the ecological science community is actively beginning to use those data to answer fundamental questions regarding the dynamical Earth system. More specifically, large-scale ecosystem modeling has been extensively used to simulate how the earth system responds to change in climate and atmospheric composition. An important class of models has been developed to reproduce the essential dynamics governing the flux exchange of energy, momentum and mass between the Earth surface and atmosphere. Special attention is required to characterize carbon dioxide mass exchange dynamics because it is one of the driving parameters of climate change. All models aiming at simulating the exchange of carbon dioxide between the atmosphere and the terrestrial surface include Leaf Area Index (LAI, i.e., the amount of one-sided green leaf per unit area) and fraction of Photosynthetically Active Radiation (fPAR) as state variables [Sellers, Ref 30]. Those parameters can be estimated using remote sensing data and in
this respect, models capable of capturing the essential physics underlying the interaction between light and vegetation play a central role in testing the efficiency of remote sensing retrieval.

As a general fact, we can say that modeling plays a pivotal role in various types of investigations connected to the use of remote sensing data. Reflectance modeling and the computation of the radiant energy distribution in vegetation canopy are important for several reasons. Reliable estimation of fPAR can be done only if accurate knowledge of the radiative regime inside the canopy is available while the radiation reflected by the canopy is generally needed for any type of remote sensing applications. Moreover the reflectance (i.e., fraction of energy reflected by the canopy) modeling is a fundamental component of scientific investigation because, when applied as a diagnostic tool, it bridges investigative conjecture with spectral data collected at laboratory or field level.

It is in considering these reasons that Ganapol at al. [Ganapol Ref 14, 15] developed a 1-D coupled Leaf-Canopy model (LCM2) capable of computing, at any given wavelength, the radiation intensity inside the canopy as well as the hemispherical reflectance at the “Top of the Canopy” (TOC). LCM2 is the result of connecting two radiative transfer models which are coupled to generate vegetation canopy reflectance as function of leaf chemistry, leaf morphology (structure), leaf thickness, soil reflectance and canopy architecture. The first model, called LEAFMOD, simulates the radiative transfer within the leaf to extract the optical properties pertaining to a specific leaf species. The second model, called CANMOD, simulates the radiative transfer at canopy level and uses the optical information coming from LEAFMOD.
Although LCM2 is unique in its feature because it is the only canopy radiative transfer model to embed the biochemical component at leaf level, it has, as any other model, its limitation and it is subject to improvement. For example it is able to predict the radiant intensity as function of the depth within the canopy and the inclination angle. In essence, it describes the photon interaction within the canopy by averaging out the effect of the azimuthal angle. While this procedure allows the problem to be simplified by reducing the number of independent variables, the effect of the azimuth is lost. For example, the sunlight fundamentally thought of as a beam of photons entering the canopy, will generally have an azimuthal angle as well as an inclination angle. Therefore, the photons will experience a scattering in the azimuth direction as well. It is evident that the incorporation of the azimuthal dependence in the LCM2 coupled leaf-canopy model is the next step to be taken to extend the capability in computing the radiative regime inside the canopy.

1.2. Radiative transfer model in dense plant canopies and Neural Networks: Inverse problem and applications

Vegetation radiative transport models, such as LCM2, usually work in the forward mode, i.e., given the leaf/canopy chemistry and structural parameters, reflectance at the top of the canopy is determined. The inverse problem is the most important for practical matters in satellite remote sensing because it allows the estimation of biophysical canopy parameters using information collected by remote satellite and airborne sensors. There are different techniques that can be used to solve the inverse problem. Look-up tables and
Quasi-Newton algorithm are widely used to approach the inversion. One other possible and definitely attractive technique is to model inversion via Neural Networks (NNs). NNs are broadly used in remote sensing from classification to inverse problem solution [Smith Ref 32, Abdelgadir Ref 1]. Biologically inspired, NNs are made of elements, called “neurons”, connected in a variety of ways depending upon the problem to be solved. Given a set containing input/output pairs representative of a continuous function, NNs can be trained to “learn” the underlying relationship. Previous works used training sets coming from different radiative transfer models to solve both direct and inverse problems [Combal Ref 6].

LCM2, or any improved version of the model, can be inverted using the neural network technique. After generating the training set by running the model in the forward mode to collect the input/output pairs representing the relationship between the reflectance and the canopy parameters to be retrieved, a neural network can be designed and trained to learn the inverse map. The designed algorithm is an extremely useful tool for remote sensing applications. For example data coming from satellites and/or airborne sensors can be processed by the neural network to retrieve information about the down-looking vegetation field.

One of the most interesting applications, which is seen as the ideal candidate for the neural network inverse procedure, is the NASA UAV Coffee Project [Herwitz Ref 19]. Over the past few years, NASA has shown tremendous interest in exploring the feasibility of deploying Unmanned Aerial Vehicles (UAVs) as long duration platforms for agricultural applications. Solar powered and remotely controlled, UAVs (e.g. Helios,
PathFinder+ from AeroVironment Inc.) can be implemented as remote sensing platforms if any hyper-spectral camera is mounted on-board for crop field monitoring. In October 2002, a group at the Ecosystem Science and Technology branch (NASA Ames) successfully flew Helios over the Kauai coffee plantation field (Hawaii). The goal was to demonstrate the technological feasibility of using these aircrafts to help growers define the best harvesting strategy [Herwitz Ref 19]. We discuss about this project more extensively in section 1.3.

LCM2 can also be modified to include other elements (e.g. coffee cherries) as absorbing/scattering elements. Its neural network inversion then produces algorithms for estimating the percentage of ripe coffee cherries. Such algorithms could be implemented on the on-board UAV micro-processor for real-time, accurate ripeness predictions.

Comprehensively, this shows that the modeling and computation of the radiative regime inside the canopy is essential for remote sensing applications. The inversion of the mathematical formulation provides a useful tool for canopy morphology as well as canopy biochemistry retrieval. Neural network algorithms that learned the inverse map generated by the canopy model have the advantage of being portable and can be implemented with any sensing devise for real-time prediction.

1.3. NASA “Coffee Project”: Searching for heavenly coffee

The “coffee project” is one of the most interesting projects funded by NASA Earth Science Enterprise (ESE). It is a large scale project which aims to demonstrate the economic potential of UAVs as long-duration platforms equipped with high resolution,
real-time imaging systems serving the private business interest in tropical agriculture. Figure 1 shows two examples of UAVs. The leftmost image is a picture of Helios while flying in the Hawaiian sky. It is an unmanned, solar powered, propeller-driven airplane manufactured by Aeroviroment Inc. (Simi Valley, CA). In August 2001, the Helios prototype became the highest flying propeller-driven aircraft in the world, reaching 96,000 ft altitude during a flight based at the Pacific Missile Range Facility (PMRF located in Kauai, Hawaii). The major attractions of the solar powered UAV are the capability of performing slow-flight speed and the high degree of aerodynamic stability. The rightmost image in Figure 1 is an image representing Helios’ smaller brother, Pathfinder Plus. It is propeller-driven and also uses solar energy as power source. It is able to achieve a flight speed as low as 30 mph. Both UAVs are able to perform long-duration flight. Extended airborne loitering ability provides an excellent opportunity to wait for clear sky conditions, therefore increasing the probability of cloud-free image acquisition, while the slow flight speed characteristics give the opportunity to acquire higher quality multi-spectral digital images. The loitering capability makes those UAVs ideal candidates for showing how remote sensing technology can be used in helping the agricultural private sector. Indeed, in the tropical sector of agriculture, coffee is a high-value crop and private companies in the coffee business always look for new methods to increase the harvesting efficiency. The latter will definitely be translated into increased revenue. UAVs can help improve the efficiency by acquiring high-resolution digital imagery in near-real time to assist coffee growers in their mechanical harvest operation, providing critically important information about the percentage of ripe cherries present in
the field. The imagery can be processed on-site in a specifically designed harvest operation office and used as a decision-making tool for selecting the best harvesting strategy. The former considerations concerning both UAVs and ripeness strategy constitute the backbone of what is now called “NASA Coffee Project”.

In June 2000, a NASA Ames-based team, leaded by Dr. Herwitz of Clark University, responded to NASA Earth Science Enterprise (ESE) research announcement that sought projects capable of demonstrating the capabilities of UAVs for earth science and commercial applications, proposing the use of UAV’s imaging capabilities as a tool for helping growers to find the best harvesting strategy. In May 2001 the proposal was awarded a grant of $3.76M to test the proposed technology.

The coffee project included the University of Arizona as pivotal player. Our Canopy Radiative Transfer Group is responsible for devising an “intelligent” algorithm capable of analyzing airborne images and estimating the percentage of ripe, over-ripe and under-ripe cherries present in the down-looking field. As previously mentioned, we proposed a model-based neural network as solution to the problem. Figure 2 shows a possible conceptual scheme for a generic harvesting session. The UAV flies over the coffee plantation field and takes images of the down-looking sector. The images are processed via a neural network for ripeness prediction. This information is sent to the ground via telemetry and made readily available to the growers who will send the appropriate signal to the mechanical harvesting devices.

In August 2002, the NASA team moved to Kauai (Hawaii) to start the Helios deployment. The place chosen for the demonstration was the coffee plantation field
owned by the Kauai Coffee Company Inc. The place was selected mainly because of its close proximity to the PMRF. Indeed, the latter provided the protected airspace needed for take-off and landing and the ground-based facility support. Another factor influencing the selection of the Kauai Coffee Company was their large scale production and the extensive use of mechanical harvesting. Moreover, the company was willing to participate in the project.

Figure 1: Helios and Pathfinder Plus flying in the Hawaiian sky

Helios was equipped with two high-resolution digital cameras to provide the desired imagery. The Kodak Pro Back was attached to a Hasselblad camera body to provide images in the visible. Its main function is to determine cloud-free spots in the sky. The second on-board camera was a DuncanTech digital multi-spectral camera equipped with
internal filters to provide narrow-band imagery in the visible and the infrared part of the spectrum.

In the early October 2002, Helios successfully flew over the Kauai coffee plantation. It was able to supply the ground station with cloud-free multi-bands images of the plantation field. A collage of field images was built to perform the analysis of the ripeness level.

Although the successful mission was a major breakthrough for the overall concept, it will take some times before this kind of technology is regularly used, the range being somewhere between 5 and 10 years. The key milestone is to achieve a more efficient solar energy storage so that these UAVs can stay aloft overnight. The idea is to store energy during the day and maintain the flight throughout the night. Finally, when this UAV technology becomes mature, unmanned airplanes should be able to stay aloft for periods of several weeks.

UAVs equipped with off-the-shelf camera systems might have other agricultural applications. Once the imaging system of the solar-powered UAV has been tested for ripeness estimation, other featured crops on tropical scenarios could benefit from similar technology. The Hawaiian island for example is a land rich of papaya, macadamia crops, flower fields, mangoes and guava. The amount of those crops could be estimated using UAVs imaging system, in a similar fashion. As result, the use of high resolution, cloud-free airborne imagery could potentially affect the way agriculture, in the broadest sense is thought and done.
Figure 2: Conceptual scheme for the on-board network operations. Images are taken by the camera system and processed by the neural network. The results is sent to the ground via telemetry.

1.4. Outline of the research to be presented

This research focuses on solving what is generally called the "end-to-end" problem in radiative transport in vegetation canopies. The aim is to develop a methodology that starts by applying first principles to generate a coupled Leaf-Canopy model and ends by devising a tool for remote sensing applications: after the numerical solutions of the model equations are found and tested, a model inversion is performed via neural network algorithms. The Neural Network inversion will provide the ultimate "intelligent" algorithm which represents the end product of the overall procedure. The neural network algorithm will be tested on images coming from the Kauai campaign (NASA UAV Coffee project) to verify the efficacy of the approach. A theoretical analysis of the canopy radiative transport equation will be done to provide major insight for further
semi-analytical and numerical solutions. The scheme presented in Figure 3 is an illustration of the five steps taken in devising the abovementioned "intelligent" algorithm, ready to be used by the growers as tool for ripeness prediction. The five steps are the subjects of the next sections.

Figure 3: From modeling to application: Solution scheme for the "End-to-End" problem
2. PART I: MODELING THE RADIATIVE REGIME IN DENSE CANOPY STRUCTURES

2.1. Introduction

This section deals with the generation of a true transport model to simulate the radiative regime within vegetation canopies. It represents the first step toward the solution of what we previously called the "end-to-end" problem. The aim is to describe the process that leads to the formulation of the mathematical equations representing the appropriate model describing the radiant intensity within a canopy structure. When numerically integrated, the model is used to compute the radiant intensity within and at the top of the canopy.

The model is based on first principles which describe the physics of interaction between light and vegetation and it improves the previously discussed LCM2 model. Indeed, the final goal will be to extend LCM2 predictive capabilities by implementing new features (i.e., incorporating azimuthal dependence of the radiant intensity as will be shown later).

We begin by considering the problem from a general point of view. The sunlight is composed of photons traveling at various wavelengths throughout the atmosphere. At the end of their journey toward the earth's surface, photons enter the top of the canopy where the interaction with the vegetation elements begins. In general, photons can be either absorbed or scattered by the canopy leaves and a closer look at the single photon's path shows that there are two different levels of interaction. At the leaf level, photons enter the
single leaf, being absorbed and scattered within the leaf's structure. At the canopy level, photons travel between leaves and experience both scattering and absorption. The above considerations suggest the generation of two different models, each appropriate for the considered level of interaction. The combination of both models will create a nested leaf-canopy transport model.

We distinguish three possible levels of description: level I, which deals with modeling the radiative regime within the leaf, level II, which describes the radiative regime within the canopy and level III, which shows how the two levels are synergistically integrated to generate the final canopy radiant energy distribution. The three levels of descriptions are the subject of the next sections.

2.2. Level One: Modeling the radiative regime within leaf (LEAFMOD derivate)

The first level of description deals with modeling the radiant energy distribution within the leaf. If we look at the overall canopy, we understand that the leaf is the fundamental constituent of the vegetation architecture and, from a radiative transfer point of view, its optical behavior, i.e., how light is scattered and/or absorbed, must be addressed. Indeed, it is well known that the leaf is the primary energy harvesting element promoting regulation and adapting control of evapotranspiration. Leaves usually intercept and absorb photons, storing energy, measured by fPAR (fraction of Photosynthetically Active Radiation), to drive carbon fixation. Moreover, the amount of various biochemical elements present in leaves plays a pivotal role in driving the carbon cycle and gas exchange with the atmosphere. Both fPAR and the foliar biochemical composition can be
estimated by using airborne and/or satellite sensors. The use of remote sensors must be coupled with appropriate radiative transfer modeling for data analysis and interpretation. The leaf radiative transfer model plays its part by partially filling the need of understanding the fundamental leaf radiative transfer phenomena.

The approaches usually taken in modeling the radiative transport in leaf structure can be divided into two categories: Probabilistic and deterministic. We will follow the deterministic approach which consists in formulating an integro-differential equation derived from the application of basic laws of physics, i.e., balance of photons in the appropriate phase-space. In a more general sense, we adopt a modeling philosophy based on two fundamental facts. First, we give preference to simplicity versus complexity. Simpler models are generally faster and also computationally cheaper. Second, we assume the model to be generally “ignorant”: The leaf interior is an extremely rich structure and any attempt to capture all possible details is doomed to fail mainly because of lack of information. Nevertheless, we conjecture that complex interaction photon-leaf interior tend to average due to the random nature of the interior cell wall structure. The simplest behavior is assumed to be a natural consequence.

Any deterministic transport model aims to describe the transport of particles in a host medium. Leaf radiative transport models describe how photons move in the leaf interior which is, as matter of fact, the host medium. In classical transport theory, the medium is assumed to be a collection of spherical scattering and absorbing centers uniformly distributed in a differential volume. The particles behavior is determined by the probability of scattering and/or absorption with the host medium, assuming that between
interactions, streaming is possible. Photons behavior can be described in this fashion and, if conservation of photons is applied in the six-dimensional phase space including space and velocity, the following equation is derived:

\[ \frac{1}{c} \frac{\partial I_\lambda(\vec{r}, \vec{\Omega})}{\partial t} + \vec{\Omega} \cdot \nabla I_\lambda(\vec{r}, \vec{\Omega}) + \Sigma_{tot}(\vec{r}, \vec{\Omega}) I_\lambda(\vec{r}, \vec{\Omega}) = \int d\Omega' \Sigma_\lambda(\Omega' \rightarrow \Omega) I_\lambda(\vec{r}, \vec{\Omega'}) \]  

(2.1)

Here \( I_\lambda(\vec{r}, \vec{\Omega}) \) is spectral the radiance (Energy/m²·ster·sec) of photons at the location \( \vec{r} \) traveling in the direction \( \vec{\Omega} \) within the cone of angles \( d\Omega \) and it is the unknown desired quantity. The two terms at the left hand side of the equation (2.1) represent net energy loss of photons streaming out of the phase space which is balanced by the energy loss due to the scattering and absorption (third term) and the inscattering of photons in the phase space (right hand side). The leaf medium is described by the absorption and scattering coefficients: \( \Sigma_{tot} \) is the total interaction coefficient and it is the sum of the absorption and scattering coefficient \( (\Sigma_{tot} = \Sigma_a + \Sigma_s) \). \( \Sigma_\lambda(\Omega' \rightarrow \Omega) \) is the differential scattering coefficient representing the probability that photons traveling in the \( \vec{\Omega}' \) direction are scattering in the \( d\Omega \) about \( \vec{\Omega} \) direction.

Equation (2.1) is a fairly complex equation. From a mathematical point of view it is known as linearized Boltzmann equation and, in the presented form, it does not have any analytical solution representation. Moreover, the scattering and absorption coefficients must be described to properly account for the medium. Although (2.1) is the basis for our modeling, several assumptions must be considered to derive a more manageable form.
First, the problem is considered time-independent. The $1/c$ term factoring the time-derivative of $I_s(\vec{r}, \vec{\Omega})$ is higher than the time-rate; therefore, it is reasonable to assume the problem is stationary. Second, we devise a one-dimensional form of equation (2.1) by considering the leaf infinite in the transverse direction (Figure 4). Next, we illustrate how to model the scattering process so that the leaf medium can be taken into account. Scattering is responsible for deflecting the photons' trajectory and it plays a fundamental role in any radiative transfer model. Usually the scattering coefficient is the most difficult to properly characterize. The photons deflection is primarily caused by discontinuities between indices of refraction at cell walls. On average, we assume nearly-random orientation of the cell walls so that isotropic scattering (equal probability for a photon to be scattered in any direction) can be safely considered a good approximation (Figure 5). Moreover, the interior of the leaf will be modeled as a homogenous mixture of biochemical elements which yields to a constant absorption coefficient throughout the overall leaf thickness. The equation (2.1) can be simplified to include the aforementioned assumptions. The simplifying assumptions generate two possible formulations: one-angle and two-angle problems. The following sections describe the radiative transfer equation in greater detail.

In the rest of the chapter, the wavelength dependence of the radiant intensity is omitted.
2.2.1. LEAFMOD model description: One-angle radiative transfer equation

The basic equation describing the balance of photons, one-dimension in space, azimuthally integrated and inclusive of all the assumptions discussed in the past section, is the following:
Where \( I(\tau, \mu) \) is the radiance of photons traveling in the \( \mu \) direction (which is called inclination), in the range \( d\mu \) and in the differential volume \( dz \) of unit transverse area. The phase function describing the photon’s scattering behavior is set to \( \frac{1}{2} \) which corresponds to isotropic scattering. It is standard practice to perform the substitution \( \tau = \Sigma z \) and express the equation in terms of optical path length \( \tau \) measured from the adaxial leaf surface. The resulting equation is:

\[
\left[ \mu \frac{\partial }{\partial \tau} + 1 \right] I(\tau, \mu) = \frac{\omega}{2} \int_{-1}^{1} d\mu' I(\tau, \mu')
\]

We also define the single scattering albedo to be \( \omega = \frac{\Sigma_s}{\Sigma} \).

Boundary conditions must be included. We assume a beam source of strength \( S_0 \) illuminates the entire adaxial surface in the direction \( \mu_0 \) while non-reentrant conditions are assumed at the abaxial and adaxial surfaces of the leaf. If we set \( \Delta = \Sigma d \), the boundary conditions can be written as follows:
While the knowledge of the radiative regime within the leaf is of general interest, for our purpose the primary quantities of concern are the hemispherical reflectance and transmittance to be defined as follow:

\[
R_f = \frac{1}{\mu_0} \int_0^1 d\mu' \mu' I(0, -\mu')
\]

\[
T_f = \frac{1}{\mu_0} \int_0^1 d\mu' \mu' I(\Delta, \mu')
\]

Hemispherical reflectance and transmittance are the leaf optical properties. The information regarding how light is reflected and transmitted by a single leaf are the input parameters of canopy radiative transport model.

2.2.2. LEAFMOD two-angle model: Governing equation

The one-angle transport equation describing the radiative transfer within the leaf can be extended to include the effect of the azimuth. The balance of photons within the space phase, leads to a modified version of the equation (2.3):

\[
\left[ \mu \frac{\partial}{\partial \tau} + 1 \right] I(\tau, \Omega) = \frac{\omega}{2} \int_0^\pi d\Omega' I(\tau, \Omega')
\]
If no average (integration) in the azimuth is performed, the intensity depends on two angles $\Omega = [\mu, \varphi]$. While interacting with the leaf interior, photons are now allowed to be scattered in any possible direction. The scattering is still assumed to be isotropic. The boundary conditions are now going to include the possibility, for the illuminating beam to assume any potential direction both in inclination and azimuth:

\[
\begin{align*}
I(0, \mu, \varphi) &= S_0 \delta(\mu - \mu_0) \delta(\varphi - \varphi_0), \mu > 0 \\
I(\Delta, \mu, \varphi) &= 0, \mu < 0
\end{align*}
\] (2.7)

Non-reentrant flux conditions are also assumed for the rightmost surface of the slab representing the leaf.

2.3. Level Two: Modeling the radiative regime within the canopy (Two-angle CANMOD)

The second and consequent level of description deals with modeling the radiative transfer regime within canopy. In section one, the importance of such model in the area of satellite and airborne remote sensing was discussed. At this level, we are interested in describing the photons' fate as they move between leaves. As in any other deterministic radiative transport model, we apply the basic conservation of particles to determine the equation describing the distribution of the radiant energy within and at the top of the
canopy. As in the LEAFMOD case, the most general transport equation, resulting from balancing the photons within the appropriate phase-space is the equation (2.1). The latter describes particles moving in a host medium. At canopy level the medium is the ensemble of leaves: photons are absorbed and scattered by the leaves which now can be considered the scattering and absorption centers. The medium enters into the transport equation via the appropriate scattering and absorption coefficients which must be modified to account for the fact that leaves do not scatter photons with the same probability in any direction. Indeed, the canopy medium is intrinsically anisotropic and the model must reflect this. We start with describing the canopy equation in the two-angle formulation which represents an extension of the one-angle existing CANMOD. The latter will also be described in a consequent section.

Figure 6: Canopy as "turbid" medium.
2.3.1. CANMOD two-angle formulation: Model description

At the canopy level, the equation describing the radiative regime inside the vegetation is derived by applying the conservation of photons to the differential element in the phase space (Figure 6). The canopy is assumed to be a "turbid" medium with infinitesimal leaves as scattering elements having a defined normal probability distribution function. The leaf normal orientation influences the scattering which now cannot be considered isotropic. In the most general case, the phase space is a six-dimensional space (position and velocity vectors) which brings to 7 the number of independent variables if time is considered. In the current CANMOD formulation, the canopy is assumed to be an infinite transversally slab of material which leads to a one-dimensional approximation. Moreover other simplifying assumptions are considered: the scattering process yields no frequency shift, the problem is assumed stationary and the effect of the azimuth is averaged. Therefore, only one spatial coordinate (canopy depth) and one angle (inclination) are defined as independent variables. We aim to improve the model by implementing a two angle formulation. If \( z \) represents the canopy depth, the appropriate transport equation for the two angle case can be derived directly from (2.1). With the appropriate positions that we subsequently describe, the two-angle CANMOD equation can be written as following:

\[
\left[ \mu \frac{\partial}{\partial \tau} + G(\mu) \right] I(\tau, \Omega) = \frac{1}{\pi} \int_0^{2\pi} d\Omega'^{\prime} \Pi(\Omega', \Omega) I(\tau, \Omega') \tag{2.8}
\]
Here \( \tau \) is the depth within the canopy in LAI units and \( \Omega = (\mu, \phi) \) is the solid angle. The boundary conditions are:

\[
I(0, \Omega) = F^i(\Omega), \mu > 0 \\
I(\Delta, \Omega) = F^k(\Omega) = \frac{r}{\pi} \int \mu' |I(\tau, \Omega'), \mu < 0
\]  

(2.9)

The first line in equations (2.9) describes the condition at the top of the canopy and usually has two contributions coming from direct and diffuse sunlight. The second line represents the reflected radiance from the bottom surface at \( \tau = \Delta \) usually resulting from a partially Lambertian reflecting soil.

The right hand side of equation (2.8) is the inscattering term. The differential scattering law is described by the area scattering phase function to be defined as follows:

\[
\frac{1}{\pi} \Gamma(\Omega', \Omega) = \int_{\Delta^2} |\Omega \cdot \Omega_L| g_L(\Omega_L)'(\Omega', \Omega; \Omega_L)
\]  

(2.10)

Where \( \Omega_L = (\phi_L, \Theta_L) \) is the leaf normal and \( g_L(\Omega_L) \) is the Leaf Angle Distribution (LAD) function. LAD is an important parameter since it describes the canopy architecture. It can be defined as the probability that a leaf has a normal \( \Omega_L \), directed away from the top surface, in a unit solid angle about \( \Omega_L \). It is often reasonable to assume that polar and azimuthal angles of the leaf are independent, i.e. we can
write \( g_L(\Omega_L) = g_L(\mu_L) h_L(\phi_L) \). Ample experimental evidences indicate that the leaf distribution function is random in the azimuthal orientation [Shultis Ref 31], i.e. \( h_L(\phi_L) = 1 \). Therefore only \( g_L(\mu_L) \) survives. There are five common leaf angle distribution functions that are in use and they are illustrated in Table 1. Figure 7 shows \( g_L(\mu_L) \) as function of the leaf normal inclination. For example canopies that fall under planophile have leaves distributed mainly horizontally while the plagiophile have the leaves mainly inclined at 90 degree.

<table>
<thead>
<tr>
<th>LAD</th>
<th>( g_L(\omega) ), ( \omega = \cos(\theta) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planophile</td>
<td>( \frac{4\omega^2}{\pi \sqrt{1-\omega^2}} )</td>
</tr>
<tr>
<td>Erectophile</td>
<td>( \frac{4(1-\omega^2)}{\pi \sqrt{1-\omega^2}} )</td>
</tr>
<tr>
<td>Plagiophile</td>
<td>( \frac{16\omega^2(1-\omega^2)}{\pi \sqrt{1-\omega^2}} )</td>
</tr>
<tr>
<td>Extremophile</td>
<td>( \frac{(18\omega^4-16\omega^2+4)(1-\omega^2)}{\pi \sqrt{1-\omega^2}} )</td>
</tr>
<tr>
<td>Unophile (spherical)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1: Leaf Angle Distribution Function (LAD). Common canopy architectures and related \( g_L(\mu_L) = g_L(\omega) \) functions.
Figure 7: LAD as function of the leaf normal inclination angle.

The leaf phase function is assumed to be bi-Lambertian:

\[\gamma(\Omega', \Omega; \Omega_i) = \begin{cases} \frac{1}{\pi} |\Omega \cdot \Omega_i|, & (\Omega \cdot \Omega_i)(\Omega' \cdot \Omega_i) < 0 \\ \frac{1}{\pi} |\Omega \cdot \Omega_i|, & (\Omega \cdot \Omega_i)(\Omega' \cdot \Omega_i) > 0 \end{cases} \]  \hspace{1cm} (2.11)

Consider a leaf and photons traveling in the direction $\Omega'$. The photons carry energy which can be absorbed, transmitted and reflected by the leaf. Consider the direction $\Omega_i$. 
The leaf scattering phase function is the ratio of energy going out in a direction \( \overline{\Omega} \) and the incoming energy in direction \( \overline{\Omega}' \). The assumption that any leaf in the canopy reflects and transmits energy as an ideal diffusive surface (Figure 8), allows the equation (2.11) definition. Reflectance and transmittance factors \((\rho_t, \tau_t\) respectively\) can be measured for any leaf specie.

\[ G(\mu) \text{ is the intercept function (or geometry factor) and it is related to the total interaction coefficient defined in equation (2.1). The total interaction coefficient, sometimes called extinction coefficient, is defined as the probability, per unit path length of travel, that the photon hits the leaf, i.e., the probability that a photon, while traveling a distance } ds \text{ along } \overline{\Omega} \text{ is intercepted by a leaf divided by the distance } ds. \text{ Mathematically this is defined as:} \]

\[ \Sigma(\tau, \overline{\Omega}) = G(\tau, \overline{\Omega}) k_l(z) \]  \hspace{1cm} (2.12)
Where $u_L(r)$ is the total one-sided area per unit volume and $G(r,\overline{\Omega})$ is defined as the fraction of the total leaf area (per unit volume of the canopy) that is perpendicular to $\overline{\Omega}$:

$$G(r,\overline{\Omega}) = \frac{1}{2\pi} \int_{4\pi} g(r,\overline{\Omega}_L, \overline{\Omega}_L \cdot \overline{\Omega}) \, d\overline{\Omega}_L$$  \hspace{1cm} (2.13)

In the case of random leaf distribution in the azimuth, the geometric factor depends only on the cosine of the inclination angle, i.e., $G(r,\overline{\Omega}) = G(\mu)$.

The equation (2.8) is an integro-differential equation. $I(r,\Omega)$ is the radiance which is defined as the energy per unit area per unit time per unit solid angle transported by photons traveling within the canopy. The inputs to this model are the leaf reflectance and transmittance (optical properties coming from LEAFMOD) as well as LAI, LAD, soil reflectance, sun inclination and azimuthal angle.

This equation can be numerically solved to determine the radiance exiting the top of the canopy. The most important quantity for remote sensing application is the hemispherical reflectance, defined as:

$$R_f = \frac{1}{\pi} \int_{4\pi} |\mu| I(0,\Omega), \mu < 0 \, d\Omega$$  \hspace{1cm} (2.14)
This quantity is what is captured by sensing devices. It is also important to note that equation (2.8) can be solved for different wavelengths to compute the spectral response of the canopy.

The model equation describes the radiant regime for a general canopy. By varying the input parameters we will be able to consider different canopies. For example, we can change the leaf angle distribution and/or the biochemical concentration to simulate a specific plant species. For the purpose of the present study, we are interested in dealing with a canopy containing coffee cherries as well as leaves, both assumed scattering and absorbing elements. This requires a modification of the canopy optical properties which can be done by defining a certain percentage of ripe, over-ripe, under-ripe cherries in the field and mixing the spectral reflectance and transmittance of the cherries with the reflectance and transmittance of the leaves coming from LEAFMOD. The canopy equation will sense this effect as a different value of reflectance and transmittance coefficients.

2.3.2. CANMOD one-angle formulation: Summary

CANMOD one-angle formulation is obtained by azimuthally averaging equation (2.8). The averaging process yields:

\[
\left[ \mu \frac{\partial}{\partial \tau} + G(\mu) \right] I(\tau, \mu) = \int_{-1}^{1} d\mu' I(\tau, \mu') (\mu', \mu) \]

(2.15)
The associated averaged boundary conditions are:

\[ I(0, \mu) = \delta(\mu - \mu_0), \mu > 0 \]
\[ I(\Delta, -\mu) = 2r \int_{-1}^{0} d\mu' |\mu'| f(\tau, \mu') \]  \hspace{1cm} (2.16)

For the one-angle CANMOD situation, it is always assumed that \( g_\ell(\overline{\Omega}_\ell) = g_\ell(\mu_\ell) h_\ell(\phi_\ell) \).

With this, the intercept function can be written as:

\[ G(\mu) = \int d\mu_g g_\ell(\mu_\ell) \psi(\mu, \mu_\ell) \]  \hspace{1cm} (2.17)

where

\[ \psi(\mu, \mu_\ell) = \frac{1}{2\pi} \int_0^{2\pi} d\phi_\ell |\overline{\Omega} \cdot \overline{\Omega}_\ell| h(\phi_\ell) \]  \hspace{1cm} (2.18)

If \( h_\ell(\phi_\ell) = 1 \) (see previous section), then (2.18) can be exactly integrated to obtain an analytical form:

\[ \psi(\mu, \mu_\ell) = \begin{cases} \mu\mu_\ell, & \text{if } \cos(\theta) < 0 \\ \mu\mu_\ell \left[ \frac{2\phi_\ell(\mu)}{\pi} - 1 \right] + \frac{2}{\pi} \sqrt{1 - \mu^2} \sqrt{1 - \mu'^2} \sin \phi_\ell(\mu), & \text{otherwise} \end{cases} \]  \hspace{1cm} (2.19)
Where

\[ \phi_i(\mu) = \cos^{-1}(\frac{1}{\ctn(\theta)\ctn(\phi_l)}) \]
\[ \theta = \cos^{-1}(\mu) \]  

(2.20)

The azimuthally averaged scattering area function becomes:

\[ \Gamma(\mu', \mu) = \int_0^1 d\mu_L g_L(\mu_L) \sigma(\mu', \mu_L) b(\mu, \mu_L) \]
\[ a(\mu', \mu_L) = \tau_L H(\mu, \mu_L) + \rho_L H(-\mu, \mu_L) \]
\[ b(\mu, \mu_L) = 2H(\mu, \mu_L) \]
\[ H(\mu, \mu_L) = \frac{1}{2\pi} \int_{\Omega} d\Omega \cdot \Omega_L \]  

(2.21)

The function \( H(\mu, \mu_L) \) can be expressed in analytical form:

\[ H(\mu, \mu_L) = \begin{cases} 
\frac{1}{\pi} \left[ \mu_L \phi(\mu_L) \sqrt{1 - \mu^2} \sqrt{1 - \mu_L^2} \sin \phi_L(\mu) \right] & \text{otherwise} \\
1 & \text{if } |\mu_L| > 1 \\
0 & \text{if } |\mu_L| < 1 
\end{cases} \]

(2.22)

The reflectance, for any desired wavelength, is computed integrating the radiant intensity at the top of the canopy:
2.4. Level Three: Coupling Leaf and Canopy model (LCM2)

The two previous portrayed levels of description can be connected to generate the Leaf-Canopy radiative transfer Model (LCM2). LEAFMOD and CANMOD have been coupled to generate vegetation hemispherical reflectance as function of leaf chemistry, leaf morphology, leaf thickness, soil reflectance and canopy architecture. This section illustrates how the coupling is performed.

Figure 9 shows the LCM2 flowchart. The leaf and canopy models have been implemented in modules for rapid and efficient connection. Moreover, a LEAFMOD inversion module was previously designed and proposed to determine the scattering and absorption coefficients associated with the single-specie leaf present in the canopy. Indeed, as previously mentioned, the connection between the two levels of descriptions occurs by the specification of the leaf optical properties, i.e., leaf reflectance and transmittance. CANMOD accept those two quantities as input and use them to determine the leaf scattering area function as prescribed by the diffusive bi-Lambertian model. For the specific canopy under consideration, one can retrieve, using experimental
information, the scattering and absorption coefficients and use them to determine, for any
given wavelength, the leaf optical properties. The latter are input to the CANMOD for the
hemispherical reflectance computation. LCM2 modular structure makes the algorithm
extremely flexible. Next, we provide a detailed description of LCM2 flowchart.

The computation starts by defining the canopy under consideration, i.e., specifying all
possible input parameters. Besides canopy architecture (LAD), LAI, soil reflectance and
sun angle, we can also specify leaf biochemistry and leaf thickness. More specifically, the
water fraction, chlorophyll, lignin plus cellulose and protein area concentration (mass/leaf
area) are provided. The first step in the LCM2 operation consists of determining the leaf
optical properties. The LEAFMOD forward module accepts leaf scattering and absorbing
coefficients and computes the leaf reflectance and transmittance. Nevertheless, the
scattering and absorption parameters are not known and must be determined. The
scattering coefficient is assumed to be mainly influenced by the leaf anatomical structure
(cell walls orientation and discontinuities in index of refraction) while the absorption
coefficient is mainly driven by the biochemistry of the leaf. Although some leaf
scattering models have been a-priori postulated, a general scattering theory is impossible
to construct due to the extreme complexity of the leaf interior. To overcome this problem,
the idea is to invert LEAFMOD and let the inverse module compute the scattering
coefficient using experimental data available in the literature. Indeed, the LCM2
computation starts by opening the LOPEX archive which is database containing optical
information for about 200 leaf species. For a specific canopy leaf under consideration, the
LOPEX archive provides the measured reflectance and transmittance for a reference leaf.
Thickness must also be specified as input parameter. This information is provided to LEAFMOD inversion module which computes scattering and absorption coefficients. To a first approximation, we assume that similar leaf species have similar interior structure and therefore share the same scattering coefficient. The computed absorption coefficient is discarded since a new one is constructed using the biochemical information for the specific leaf of interest. The assumption that the biochemistry mainly influences the absorption coefficient allows us to construct it for the specified leaf of the canopy. The prescription is the following:

\[ \Sigma_a = \sum_{i=1}^{n} \rho_i \sigma_i \] (2.24)

Here, \(\sigma_i\) is the specific absorptivity and \(\rho_i\) is the concentration for any specified biochemical agent.

The scattering and absorption coefficients are therefore processed by the LEAFMOD forward module to determine reflectance and transmittance of the specific leaf under consideration.

Reflectance and transmittance are inputs for the CANMOD module which also accepts LAI, LAD, soil reflectance and sun angle. CANMOD outputs, at any specified wavelength, the hemispherical reflectance at the top of the canopy.

The next two sections discuss the theoretical investigation and numerical simulation of the leaf and canopy radiative transport model presented above.
Figure 9: LCM2 flow chart. Connection between LEAFMOD and CANMOD
3. PART II: SPECTRAL THEORY FOR THE CANOPY EQUATION

3.1. Introduction

Modeling the radiative regime within dense vegetation canopies yields two integro-differential equations. For both transport in leaves and canopies, the correct application of the conservation of photons allowed us to derive two linear Boltzmann equations. More specifically, the canopy equation was presented in both a one-angle and two-angle formulation. In this section, we will focus on the canopy case, one-angle formulation. The goal is to analyze the spectral structure of the operator arising from the physical situation describing the interaction photons-canopy and put the results in the frame of a consistent spectral theory for the canopy equation.

Indeed, the canopy transport equation can be classified as a linear transport equation which has been extensively studied and applied in various fields such as neutron transport theory, radiative stellar transport and kinetic theory of gases. In its one-speed, 1-D, infinite medium, stationary form, the canopy equation can be investigated using the so-called "Singular Eigenfunctions Expansion" (SEE) method, also called "Caseology" after Kenneth Case, who was one of the founders of the technique. The method of SEE is modeled after the Fourier approach to partial differential equations. The basic idea is to construct a complete set of eigenfunctions which are derived from the solutions of the homogeneous transport equation with separated variables. The modes are used to expand
arbitrary solutions and the main task consists of the determination of the expansion coefficients.

Historically, these ideas were first formulated by Davidson [Ref 8] who also worked on some basic proofs. Other independent proposals were made by Wigner [Ref 35] and Van Kampen [Ref 33, 34]. However, in the 1960, Case [Ref 4] successfully demonstrated the power and usefulness of the method. After 1960 the method generated a large amount of interest. It was used to solve many other problems and it was extended to more complicated forms of the equation. Most of this work can be found in Case and Zweifel book [Ref 38] and in a Zweifel review paper [Ref 37]. More recently, the method was discussed in books such as Bell and Glasstone [Ref 2] and Williams [Ref 36].

The SEE method is also refereed as an "exact" method of solution for the transport equation. It is obviously intended to produce results in closed analytical form for the simple cases while in more complicated situations, it carries a fraction of the work analytically. The primary usefulness of the method lies in understanding the mathematical structure and the behavior of the solutions of the transport equation. Such knowledge is a prerequisite before attacking realistic problems using numerical methods such as discrete ordinate techniques. Closed forms solutions also provide a mean for testing the accuracy achieved by numerical approximations.

In the rest of this section, we will apply the SEE method to the one-angle canopy transport equation. We will determine the spectrum and the eigenfunctions of the canopy transport operator. Finally we will prove orthogonality and completeness.
3.2. Spectral theory for canopies with separable kernel: Formulation

In the previous section, we derived two integro-differential equations describing the photons transport in both leaves and canopy media. The equations arising in both situations are special cases of the most general transport equation usually known as linearized Boltzmann equation. In the LEAFMOD case, the uniform absorption and isotropic scattering assumptions yield an equation which is very similar to the stationary, one dimensional transport of neutrons in nuclear reactors. Although the equation is interesting from a spectral point of view, it has been extensively studied and the spectrum is very well defined. In the CANMOD case, the scattering is typically anisotropic due to the turbid medium assumption required to describe the interaction of light and foliage within the canopy structure. The area scattering function takes into account the inscattering phenomenon and its structure is extremely complex as indicated by equation (2.10) for the two-angle formulation or equation (2.15) for the one-angle case. Indeed, the canopy equation is the subject of our spectral investigation. The general equation, in the two angle case, is too difficult to analyze using the analytical tools required to apply the singular eigenfunctions method. Therefore, some assumptions are desired to reduce the level of complexity and derive an equation in a more manageable form.

We start by considering the simpler canopy transport equation and its one angle formulation:

\[
\left[ \mu \frac{\partial}{\partial \tau} + G(\mu) \right] f(\tau, \mu, \mu^*) = \int_{-1}^{1} d\mu T(\mu', \mu) f(\tau, \mu', \mu^*)
\]

(3.1)
Here the intercept function $G(\mu)$ and the area scattering function $\Gamma(\mu', \mu)$ have the usual structure outlined by equations (2.17)-(2.22). The $\mu'$ symbol indicates that the canopy is illuminated by a photon beam incident the top of the vegetation with a constant inclination angle. A closer look at the aforementioned equations highlights the complications introduced into the canopy equation by those two parameters. The area scattering depends on the leaf phase function $\gamma(\Omega', \Omega; \Omega_{z})$ as well as on the leaf normal distribution function which, in the one-angle formulation can be thought only as a function of the leaf inclination angle. To simplify the formulation, we assume that the leaves can be distributed according to the following prescription:

$$g_{l}(\mu_{l}) = \sum_{n=1}^{N} \alpha_{n} \delta(\mu_{l} - \mu_{z}).$$

Moreover, the $\alpha_{n}$ are constrained by the following relationship

$$\int_{0}^{1} d\mu_{l} g_{l}(\mu_{l}) = 1 \Rightarrow \sum_{n=1}^{N} \alpha_{n} = 1$$

This prescription (3.2) is equivalent to imposing that the leaves are oriented only in a set of $N$ discrete directions. The assumption has a profound impact on both area scattering and intercept functions. Substituting (3.2) into (2.21) and (2.17) we obtain:
where

\[
\Gamma(\mu', \mu) = \sum_{n=1}^{N} \alpha_n a_n(\mu) b_n(\mu')
\] (3.4)

where

\[
a_n(\mu) = \tau_i H(\mu, \mu_n) + \rho_i H(-\mu, \mu_n)
b_n(\mu) = 2H(\mu, \mu_n)
\] (3.5)

and

\[
G(\mu) = \sum_{n=1}^{N} \alpha_n \psi(\mu, \mu_n)
\] (3.6)

where

\[
\psi(\mu, \mu_n) = \left\{ \begin{array}{ll}
\left[ \mu \mu_n, \text{if cm}(\theta) \text{cm}(\theta_i) > 1 \right] \\
\mu \mu_n \left[ \frac{2\phi_i(\mu)}{\pi} - 1 \right] + \frac{2}{\pi} \sqrt{1-\mu^2} \sqrt{1-\mu'^2} \sin \phi_i(\mu), \text{otherwise}
\end{array} \right.
\] (3.7)

The area scattering function becomes a finite summation of products of functions in \( \mu \) and \( \mu' \). Therefore, with these assumptions, the scattering kernel is separable and of finite rank.
Thus, the canopy equation is written as:

\[
\left[ \mu \frac{\partial}{\partial \tau} + G(\mu) \right] I(\tau, \mu; \mu') = \sum_{n=1}^{N} \alpha_n(\mu) \int d\mu' b_n(\mu') I(\tau, \mu', \mu') \quad (3.8)
\]

This equation is called "canopy equation with separable kernel of finite rank".

We now define a new "angular" variable which is assumed monotonic:

\[
\xi = \frac{\mu}{G(\mu)} \quad (3.9)
\]

Here, \( \xi_m = \max(\xi) \) represents the maximum value. The conservation of photons dictates that the following condition must be satisfied:

\[
I(\tau, \mu; \mu') d\mu = I(\tau, \xi; \xi') d\xi \quad (3.10)
\]

Dividing the equation (3.8) by \( G(\mu) \) and setting:

\[
\tilde{a}_n(\mu) = \frac{a_n(\mu)}{G(\mu)} \Rightarrow a_n(\xi) \quad (3.11)
\]

We can write the final form of the equation we are going to investigate:
3.3. Singular Eigenfunctions Method: Caseology for the canopy transport equation.

Equation (3.12) is the object of our spectral investigation. It is a transport integro-differential equation with finite rank and separable kernel. To find the spectrum and the eigenfunctions of the transport operator, we apply the SEE methodology. The method consists in finding the solution of the integro-differential equation using a procedure analogous to the separation of variable commonly employed in solving partial differential equation. The general approach in solving a PDE using separation of variables consists of three steps:

1. Seek separable solution in the space and angular variable

2. This will yield ordinary differential equations both in space and angle. The angular equation takes the form of an eigenvalue problem. The problem is solved by finding both eigenvalues and eigenfunctions (elementary solutions).

3. Seek the general solution of the original equation (PDE or integro-differential equation) as a combination of these eigenfunctions.
The procedure described above will be applied to find the eigenvalues and eigenfunctions of the canopy equation in the form (3.12).

3.3.1. Caseology: Separation of variables and the eigenvalue problem

We start the process noticing that the spatial part has already been found by applying separation of variable for conventional stationary, one-dimensional problems (see reference). Thus, we look for solution in the form:

\[
I(\tau, \xi, \xi') = \phi_\tau(\xi) e^{\frac{-\tau}{\nu}}
\]  

(3.13)

Substituting (3.13) into (3.12) we get:

\[
\left(1 - \frac{\xi}{\nu}\right) \phi_\tau(\xi) = \sum_{n=1}^{N} \alpha_n a_n(\xi) \int_{\xi_n}^{\xi} d\xi' b_n(\xi') \phi_\tau(\xi')
\]  

(3.14)

Define the N "moments" of \(\phi_\tau(\xi)\) in the following way:

\[
\phi_n(\nu) = \int_{-\xi_n}^{\xi_n} d\xi b_n(\xi) \phi_\tau(\xi)
\]  

(3.15)
From a geometric point of view, the moments defined by (3.15) are the projection of the solution $\phi_n(\xi)$ in the space generated by $b_n(\xi)$. After some algebraic manipulation, equation (3.14) can be written as:

$$\nu - \xi \phi_n(\xi) = \nu \sum_{n=1}^{N} \alpha_n a_n(\xi) \phi_n(\nu)$$

(3.16)

With the following positions:

$$\alpha = \text{diag}\{\alpha_1, \alpha_2, \ldots, \alpha_N\}$$

$$\phi(\nu) = [\phi_1(\nu), \phi_2(\nu), \ldots, \phi_N(\nu)]^T$$

$$a(\xi) = [a_1(\xi), a_2(\xi), \ldots, a_N(\xi)]^T$$

(3.17)

Equation (3.16) can be written in matrix form

$$\nu - \xi \phi_n(\xi) = \nu [\alpha \cdot a(\xi)]^T \phi(\nu)$$

(3.18)

Equation (3.16) or its compact form (3.18) is our eigenvalue problem in the angular variable. In contrast with what is usually found in solving PDE, the problem does not look like a regular Sturm-Liouville eigenvalue problem. Indeed, the problem is singular due to the presence of the term $(\nu - \xi)$ on the RHS of the equation.
Since $\xi \in [\xi_m, \xi_n]$, we distinguish two possible cases depending on the value assumed by $\nu$. If the case $\nu \notin [\xi_m, \xi_n]$, we are dealing with finding the discrete spectrum. If indeed $\nu \in [\xi_m, \xi_n]$, we are exploring the continuous part of the spectrum.

The following sections will show how we approached the problem. We begin with finding the discrete eigenvalues.

3.3.2. Caseology: Discrete spectrum

Assume $\nu \in [\xi_m, \xi_n]$. The discrete eigenfunctions (modes) can be found simply by dividing equation (2.16) by $\nu - \xi$. Thus:

$$\phi_n(\xi) = \frac{\nu_0}{\nu_0 - \xi} \sum_{n=1}^{N} \alpha_n a_n(\xi) \phi_n(\nu_0) = \frac{\nu_0}{\nu_0 - \xi} [\alpha \cdot a(\xi)]^T \phi(\nu_0)$$

Here $\nu_0$ represents the sought discrete eigenvalue(s). The problem is now to find the discrete eigenvalues associated with the discrete modes (2.19) and outline a scheme for finding the moments $\phi_n(\nu_0)$. To solve the problem, we start by projecting the discrete modes on the space generated by $\xi$. Integrating the product of each mode per the $N$ functions $b_n(\xi)$ we get:

$$\phi_n(\nu_0) = \int_{\xi_m}^{\xi_n} d\xi b_n(\xi) \phi_n(\xi) = \sum_{n=1}^{N} \alpha_n \nu_0 \left( \int_{\xi_m}^{\xi_n} d\xi \frac{a_n(\xi) b_n(\xi)}{\nu_0 - \xi} \right) \phi_n(\nu_0)$$

(3.20)
These relationships are a set of $N$ equations for the moments, with $k=1, 2... N$. Define the following N-by-N matrix:

$$L_{\nu_0}(\nu_0) = \int_{-\infty}^{\infty} d\xi \frac{a_s(\xi)b_i(\xi)}{\nu_0 - \xi}$$

(3.21)

Remembering that $\phi_n = \delta_{nn} \phi_n$, we can write (3.20) as follows:

$$[\delta_{\nu_n} - \nu_0 \alpha_n L_{\nu_0}(\nu_0)] \phi_n(\nu_0) = 0$$

(3.22)

We can also reformulate (3.22) in a more compact matrix form:

$$L(\nu_0) = \int_{-\infty}^{\infty} d\xi \frac{b(\xi) a(\xi)}{\nu_0 - \xi}$$

(3.23)

$$[L - \nu_0 \alpha L(\nu_0)] \phi(\nu_0) = 0$$

(3.24)

Or:

$$A(\nu_0) \phi(\nu_0) = 0$$

(3.25)
Where:

\[
\mathcal{A}(v_o) = [A_{vn}(v_o)] = \delta_{vn} - v_o \alpha_n L_{vn}(v_o) \quad (3.26)
\]

Equation (3.25) is a homogeneous system of equations in the unknowns \(\phi_n(v_o)\). The system has a non trivial solution if and only if:

\[
\det(\mathcal{A}(v_o)) = 0. \quad (3.27)
\]

Indeed, we find the discrete eigenvalues by solving (3.27). The equation is non-linear and in general we can state that there are \(M < N\) discrete eigenvalues. Once the eigenvalues are found, we solve (3.25) \(M\)-times to find the \(M\) sets of \(\phi_n(v_o)\) (\(N\) per each eigenvalue). Thus, the discrete eigenfunctions are found according to (3.19).

3.3.3. Dispersion relationship for the canopy transport equation

In this section, we will try to manipulate equation (3.27) to define the dispersion relationship. The latter function and its analytical properties will be extremely important in proving completeness.

Consider the equation (3.27). Since the determinant is present on the LHS, we can apply the 1st Lagrange theorem (cofactors expansion) to compute the determinants. We write:
Here \( f_{nk}(v_0) \) are the cofactor computed expanding the determinant using the \( k \)-th row. Indeed, according to the theorem, any of the \( k = 1, 2, ..., N \) rows can be used. Since the expressions (3.28) are equivalent, we choose the first row (\( k = 1 \)):

\[
\det(A(v_0)) = \sum_{n=1}^{N} f_{1n}(v_0) A_{1n}(v_0) = 0
\]  

(3.29)

Equation (3.29) is multiplied and divided by \( f_{11} \) to get:

\[
\det(A(v_0)) = \frac{1}{f_{11}} \sum_{n=1}^{N} f_{1n}(v_0) A_{1n}(v_0) = 0
\]  

(3.30)

From equation (3.25) we also consider the first row (\( k = 1 \)):

\[
\sum_{n=1}^{N} A_{1n}(v_0) \phi_n(v_0) = 0
\]  

(3.31)

Comparing (3.30) and (3.31) we find that
\[ \phi_n(v_0) = \frac{f_{1n}(v_0)}{f_{11}} \quad (3.32) \]

The result (3.32) can be put back into equation (3.30) together with the definition (3.26):

\[ \det(A(v_0)) = f_{11}(v_0) \sum_{n=1}^{N} \left[ \delta_{1n} - v_0 \alpha_n L_{1n}(v_0) \right] \psi_n(v_0) = 0 \quad (3.33) \]

Now, define \( \Lambda(v_0) \) to be:

\[ \Lambda(v_0) = \frac{1}{\delta_1(v_0) \sum_{n=1}^{N} \alpha_n \alpha_n(v_0) \psi_n(v_0)} \sum_{n=1}^{N} \left[ \delta_{1n} - v_0 \alpha_n L_{1n}(v_0) \right] \psi_n(v_0) = 0 \quad (3.34) \]

The function (3.34), according to the (3.33) is zero. Indeed, (3.34) is the dispersion relationship for the canopy transport equation under investigation. Equation (3.34) can also be manipulated:

\[ \Lambda(v_0) = \frac{1}{\delta_1(v_0) \sum_{n=1}^{N} \alpha_n \alpha_n(v_0) \psi_n(v_0)} \left( \phi_1(v_0) - \sum_{n=1}^{N} v_0 \alpha_n L_{1n}(v_0) \phi_n(v_0) \right) = 0 \quad (3.35) \]
We now apply the normalization condition by setting $\phi_1(\nu_0) = 1$. Moreover, we define the dispersion function in the complex plane to be the following:

$$
\Lambda(z) = \frac{1}{b_1(z) \sum_{n=1}^{N} \alpha_n a_n(z) \phi_n(z)} \left( 1 - z \sum_{n=1}^{N} \alpha_n L_{1n}(z) \phi_n(z) \right) \quad (3.36)
$$

The analytical structure of the dispersion function will play a pivotal role in proving completeness.

### 3.3.4. Caseology: Continuous spectrum

Assume now $\nu \in [-\xi_m, \xi_m]$. This situation will give us the opportunity to explore the structure of the continuous spectrum. The eigenfunctions are formally determined by dividing equation (3.16) by $\nu - \xi$. Since $\nu \in [-\xi_m, \xi_m]$, the modes are not regular functions but must be interpreted in the sense of distributions. We find that:

$$
\phi_n(\xi) = \left( P \frac{\nu}{\nu - \xi} + \lambda(\nu) \delta(\nu - \xi) \right) \sum_{n=1}^{N} \alpha_n a_n(\xi) \phi_n(\nu) \quad (3.37)
$$

If, for any $\nu \in [-\xi_m, \xi_m]$, we were able to find the function $\lambda(\nu)$, the spectrum would be continuous with $\nu \in [-\xi_m, \xi_m]$. Indeed, this is the case.
The problem is therefore to show that it is possible to find $\lambda(v)$ and in the process devise a scheme to find the moments $\phi_k(v)$.

We start by projecting the modes in the space generated by $b_n(\xi)$:

$$
\phi_k(v) = \sum_{n=1}^{N} \alpha_n v \left( \int_{-\infty}^{\infty} \frac{a_n(\xi) b_k(\xi)}{v - \xi} \right) \phi_n(v) + \lambda(v) \sum_{n=1}^{N} \alpha_n b_k(v) a_n(v) \phi_n(v) \tag{3.38}
$$

Here, we have $k, n = 1, 2 \ldots N$. The equation (3.38) can be reformulated as follows:

$$
\phi_k(v) = \sum_{n=1}^{N} \alpha_n \left[ \lambda_k(v) + \lambda(v) b_k(v) a_n(v) \right] \phi_n(v) \tag{3.39}
$$

Where

$$
L_{kn} = P \int_{-\infty}^{\infty} \frac{a_n(\xi) b_k(\xi)}{v - \xi} \tag{3.40}
$$

Moreover, remembering that $\phi_n = \delta_{kn} \phi_k$ will yield:

$$
\sum_{n=1}^{N} (\delta_{kn} - \alpha_n \left[ \lambda_k(v) + \lambda(v) b_k(v) a_n(v) \right]) \phi_n(v) = 0 \tag{3.41}
$$
In a more compact form (3.41) becomes:

\[
\begin{bmatrix}
I - \nu \alpha L(v) - \lambda(v) k b(v) d(v)^T
\end{bmatrix} \phi(v) = 0
\]

(3.42)

Finally, using equation (3.26) we write the following:

\[
\sum_{n=1}^{N} \left( A_{kn}(v) - \lambda(v) b_k(v) \alpha_n(v) \right) \phi_n(v) = 0
\]

(3.43)

\[
\begin{bmatrix}
A(v) - \lambda(v) k b(v) d(v)^T
\end{bmatrix} \phi(v) = 0
\]

(3.44)

Equation (3.43), or equivalently (3.44), represents a system of homogeneous equations in the \( \phi(v) \) unknowns. It has a non-trivial solution if and only if:

\[
\det \left( A(v) - \lambda(v) k b(v) d(v)^T \right) = 0
\]

(3.45)

To determine \( \lambda(v) \) we apply to (3.45) a procedure similar to the one used in the case of the discrete modes. Indeed, we employ the cofactor expansion theorem to re-express the determinant (3.45). For any possible rows \( (k = 1, 2, ..., N) \) we write:

\[
\sum_{n=1}^{N} f_{kn}(v) \left[ A_{kn}(v) - \lambda(v) b_k(v) \alpha_n(v) \right] = 0
\]

(3.46)
Since any row cofactor expansion gives the same determinant, we consider only the first row ($k = 1$). Moreover, we multiply and divide by $f_{11}$:

$$f_{11} \sum_{n=1}^{N} \frac{f_{1n}(\nu)}{f_{11}} \left[ A_{1n}(\nu) - \lambda(\nu) b_{1}(\nu) a_{n}(\nu) \right] = 0$$

(3.47)

From the linear equations (3.43) we take only the first row:

$$\sum_{n=1}^{N} \left( A_{1n}(\nu) - \lambda(\nu) b_{1}(\nu) a_{n}(\nu) \right) \phi_{n}(\nu) = 0$$

(3.48)

Comparing (3.47) and (3.48) we can say that:

$$\phi_{n}(\nu) = \frac{f_{1n}(\nu)}{f_{11}(\nu)}$$

(3.49)

Some manipulation of the equation (3.48) is required to obtain $\lambda(\nu)$:

$$\sum_{n=1}^{N} \left( \delta_{1n} - \nu \alpha_{n} L_{1n}(\nu) \right) \phi_{n}(\nu) = \lambda(\nu) b_{1}(\nu) \sum_{n=1}^{N} \alpha_{n} a_{n}(\nu) \phi_{n}(\nu)$$

(3.50)

After setting $\phi_{1}(\nu) = 1$, $\lambda(\nu)$ can be derived:
\[
\lambda(\nu) = \frac{1}{b_1(\nu) \sum_{n=1}^{N} \alpha_n a_n(\nu) \phi_n(\nu)} \left[ 1 - \sum_{n=1}^{N} \nu \alpha_n L_{1n}(\nu) \phi_n(\nu) \right]
\]  

(3.51)

The outlined procedure shows that for any \( \nu \in [-\xi_m, \xi_m] \), \( \lambda(\nu) \) can be found using equation (3.51). Thus, the homogenous system (3.43) (or equivalently (3.44)) has a non-trivial solution and the moments \( \phi(\nu) \) can be determined. In conclusion, any \( \nu \in [-\xi_m, \xi_m] \) is an eigenvalue and belongs to the continuous spectrum of the transport operator. The eigenfunctions are determined by (3.37).

3.3.5. Boundary values of the dispersion function

The complex dispersion function is defined by (3.36). Its analytical structure is extremely important in determining orthogonality and completeness of the modes. The major player in determining the analyticity of the dispersion function is \( L_{1n}(z) \) defined as follow:

\[
L_{1n}(z) = \int_{-\xi_n}^{\xi_n} d\xi \frac{a_n(z) \phi_1(z)}{\nu - z}
\]  

(3.52)

In our analysis, we assume that the functions \( a_n(z), b_n(z) \) are analytical in the entire complex plane. This assumption is restrictive but it greatly simplifies the problem as
shown later. With this assumption, the dispersion function is analytical in the cut plane 
\([\xi_m, \bar{\xi}_m]\).

We are now interested in determining the boundary values of the \(\Lambda(z)\) as the \(z\)-variable approaches the cut \([\xi_m, \bar{\xi}_m]\), i.e., \(\Lambda(v)^\pm\) for \(z \to v \in [\xi_m, \bar{\xi}_m]\). First, we report the well-known Plemelj formula:

\[
\lim_{z \to 0} \frac{1}{v \pm i\varepsilon - \mu} = P \frac{1}{v - \mu} \pm i\pi \delta(v - \mu) \tag{3.53}
\]

Thus, we use (3.53) to take the following limit:

\[
\Lambda(v)^\pm = \lim_{z \to \nu} \Lambda(z) \tag{3.54}
\]

After a few steps required manipulating (3.54) and (3.53), we obtain the following:

\[
\Lambda^\pm(v) = \frac{1}{b_1(v) \sum_{n=1}^{N} \alpha_n a_n(v) \phi_n(v)} \left[ 1 - \sum_{n=1}^{N} v \zeta_n L_n(v) \phi_n(v) \right] \pm i\pi v \tag{3.55}
\]

The first term on the RHS of (3.55) is, according to (3.51), the definition of \(\lambda(v)\). We finally write the desired boundary values of the dispersion function:
\[ \Lambda(\nu)^* = \lambda(\nu) \pm i\pi \nu \] (3.56)

3.3.6. Orthogonality of the discrete and continuous eigenfunctions

In this section we show that the set of discrete and continuous eigenfunctions form an orthogonal set. Indeed, we prove the following theorem:

**Theorem: Full-range orthogonality:** The full set of eigenfunctions \( \{\phi_n(\nu)\}_{n=1}^\infty \) are orthogonal in the sense that

\[ \int_{-\infty}^\infty d\xi \phi_n(\xi)\phi_{n'}(\xi) = 0, \quad n \neq n' \] (3.57)

**Proof:** Consider the eigenvalue problem (3.16) associated with the eigenvalue \( \nu \)

\[ \left( 1 - \frac{\xi}{\nu} \right) \phi_n(\xi) = \sum_{n=1}^N \alpha_n a_n(\xi) \int_{-\infty}^\infty d\xi' b_n(\xi') \phi_n(\xi') \] (3.58)

Moreover, consider the alternate form of eigenvalue problem obtained using the transpose of the inscattering term associated to the eigenvalue \( \nu' \):

\[ \left( 1 - \frac{\xi}{\nu'} \right) \phi_n(\xi) = \sum_{n=1}^N \alpha_n b_n(\xi) \int_{-\infty}^\infty d\xi' a_n(\xi') \phi_n(\xi') \] (3.59)
Multiply (3.58) by $\phi_n(\xi), (3.59)$ by $\phi_n(\xi')$, integrate and subtract. The RHS can be expressed in the following form:

$$
\sum_{n=1}^{N} \alpha_n \int_{\xi_m}^{\xi_M} d\xi \int_{\xi_m}^{\xi_M} d\xi' a_n(\xi) b_n(\xi') \phi_n(\xi') - \sum_{n=1}^{N} \alpha_n \int_{\xi_m}^{\xi_M} d\xi \int_{\xi_m}^{\xi_M} d\xi' a_n(\xi') b_n(\xi) \phi_n(\xi) \phi_n(\xi')
$$

Equation (3.60) is zero since the two terms are identical ($\xi$ and $\xi'$ can be switched). The full operation yields:

$$
\left( \frac{1}{\nu} - \frac{1}{\nu'} \right) \int_{\xi_m}^{\xi_M} d\xi \xi \phi_n(\xi) \phi_n(\xi) = 0
$$

(3.61)

If the two eigenvalues are different, i.e. $\nu \neq \nu'$, equation (3.61) yields (3.57), which complete the proof.

For each eigenvalue $\nu$ the normalization constant is defined as usual:

$$
N(\nu) = \int_{\xi_m}^{\xi_M} d\xi \xi [\phi_n(\xi)]^2
$$

(3.62)

3.3.7. Completeness of the complete set of eigenfunctions

One of the main problems arising from the application of the singular eigenfunction expansion method to the canopy equations is the problem of proving the
completeness of the entire set of eigenfunctions. We say that an infinite set of functions \( \{\phi_n(\xi)\}_{n=1}^{\infty} \) defines a complete space if any sequence, chosen from the set of interest, converges, in some suitable norm, to any function in the space. Alternatively, one can say that any function living in the space of interest can be represented by a linear combination of \( \{\phi_n(\xi)\}_{n=1}^{\infty} \). Thus, the property of completeness of the eigenfunctions is essential since any solution of the canopy transport equation can be expanded in such a set. We therefore need to prove the following theorem:

**Theorem: Full-Range Completeness:** The M discrete eigenfunctions \( \{\phi_m(\xi)\}_{m=1}^{M} \) and the continuous eigenfunctions \( \{\phi_v(\xi)\}_{v=-\infty}^{\infty} \in [-\xi_m, \xi_m] \) are complete for the class of functions \( \psi(\xi) \in H^* \) defined on the full-range \( \xi \in [-\xi_m, \xi_m] \).

**Proof:** First, we define the class of functions (solution of the transport equation) we are interested. The functions \( \psi(\xi) \) belong to the space of the Holder continuous functions \( (H^*) \). We say that \( \psi(\xi) \in H^* \), \( \xi \in [-\xi_m, \xi_m] \), if:

1. For any two points \( \xi_1, \xi_2 \) in \( [-\xi_m, \xi_m] \) there exist two positive constants A and k such that \( |\psi(\xi_1) - \psi(\xi_2)| \leq A|\xi_1 - \xi_2|^k \). If this is true, we say \( \psi(\xi) \in H \) or \( \psi(\xi) \) satisfies the Holder condition.
2. Furthermore, if near the ends points of \([-\xi_m, \xi_m]\) we have

\[\psi(\xi) = \frac{\psi(\xi)^\alpha}{(\xi - \epsilon)^\alpha}, \quad 0 \leq \alpha \leq 1\]

where \(\psi(\xi)^* \in H\), then we say that \(\psi(\xi) \in H^*\).

The approach we follow in proving the full-range completeness theorem is constructive: we will demonstrate that any function \(\psi(\xi) \in H^\star\) can be represented as a combination of discrete and continuous modes by finding the coefficients of the expansion.

We start by writing the eigenfunctions expansion of \(\psi(\xi) \in H^\star\):

\[\psi(\xi) = \sum_{m=1}^{M} d_m \phi_m(\xi) + \int_{\xi_m}^{\xi} dA(\nu) \psi(\xi)\]

We bring the discrete part to the LHS of the (3.63) and focus only on the continuous part. The coefficients of the discrete modes will be determined later:

\[\tilde{\psi}(\xi) = \psi(\xi) - \sum_{m=1}^{M} d_m \phi_m(\xi) = \int_{-\xi_m}^{\xi_m} dA(\nu) \psi(\xi)\]

Consequently, we substitute the continuous modes (3.37) into (3.68) to get:

\[\tilde{\psi}(\xi) = P \int_{-\xi_m}^{\xi_m} dA(\nu) \sum_{n=1}^{N} \alpha_n(\xi) \phi_n(\nu) + A(\xi)\lambda(\xi) \sum_{n=1}^{N} \alpha_n(\xi) \phi_n(\xi)\]
This is an inhomogeneous singular integral equation in the unknown function $A(\xi)$. One refers to such equations as “Singular Integral Equations of the Cauchy-type". The problem of completeness of the eigenfunctions has been translated in demonstrating the existence and uniqueness of the equation (3.65) for any function $\psi(\xi) \in H^+$ and determining the coefficients of the discrete modes along the way. The most direct method of dealing with these problems is to directly solve equation (3.65).

The usual technique, employed in solving these types of integral equations, consists in converting the equation into a boundary-value problem in the complex variable. To this end, we define the following complex function:

$$n(z) = \frac{1}{2\pi i} \int_{-\infty}^{\infty} dv \frac{A(v)}{v - z} \sum_{n=1}^{N} \alpha_n a_n(z) \phi_n(v)$$

(3.66)

If $A(v) \in H^+$ and the coefficients $a_n(z)$ are analytical everywhere, then the function $n(z)$ is analytical in the cut-plane $[-\xi_m, \xi_m]$. Moreover, the function behaves $O\left(\frac{1}{|z|}\right)$ as $z \to \infty$.

The Plemelj formula (3.59) can be applied to determine the behavior of the $n(z)$ as $z$ approaches to the cut-plane:

$$n^+(\xi) - n^-(\xi) = \frac{p}{2i\pi} \int_{-\infty}^{\infty} dv \frac{A(v)}{v - z} \sum_{n=1}^{N} \alpha_n a_n(z) \phi_n(v)$$

$$n^+(\xi) + n^-(\xi) = \xi A(\xi) \sum_{n=1}^{N} \alpha_n a_n(\xi) \phi_n(\xi)$$

(3.67)
The relationships (3.67) are introduced to the LHS of the integral equation (3.65) to get:

\[ \xi \tilde{\psi}(\xi) = i \pi \xi [n^+ (\xi) + n^- (\xi)] + \lambda(\xi) [n^+(\xi) - n^-(\xi)] \]  \hspace{1cm} (3.68)

After some manipulation, we obtain:

\[ \xi \tilde{\psi}(\xi) = n^+(\xi) [\lambda(\xi) + i \pi \xi] + n^- (\xi) [\lambda(\xi) - i \pi \xi] \]  \hspace{1cm} (3.69)

Using equation (3.62) and noticing that the dispersion function \( \Lambda(z) \) and \( n(z) \) are analytical on the same cut-plane \([- \xi_m, \xi_m]\) we write:

\[ \xi \tilde{\psi}(\xi) = (n(\xi) \Lambda(\xi))^* - (n(\xi) \Lambda(\xi))^* = X^+(\xi) - X^- (\xi) \]  \hspace{1cm} (3.70)

Equation (3.70) is usually known as “Inhomogeneous Hilbert Problem”. From the above equation we can determine the function \( n(z) \). Indeed, if \( \xi \tilde{\psi}(\xi) \in H^* \) and \( n(z) \) is of degree not greater than \( k \), then applying the Plemelj formula, one obtains:

\[ n(z) = \frac{1}{\Lambda(z)} \left[ \frac{1}{2\pi i} \int_{\xi_m}^{\xi_m} d\xi \frac{\xi \tilde{\psi}(\xi)}{\xi - z} + P_k(z) \right] \]  \hspace{1cm} (3.71)
Where $P_i(z)$ is a polynomial of degree not greater than $k$. The polynomial must be determined. We notice that the function $n(z)$ is of order $O(\sqrt{z})$ as $z \to \infty$; thus, $P_i(z)$ must be zero to satisfy the previous condition.

Once $n(z)$ is found, we can determine $A(\nu)$ by simply applying the second of the (3.67):

$$A(\xi) = \frac{n^+(\xi) + n^-(\xi)}{\xi \sum_{n=1}^{N} \alpha_n a_n(\xi) \phi_n(\xi)}$$

(3.72)

And the problem is basically solved.

The last issue that must be addressed is associated to the determination of the coefficients of the discrete modes in the expansion (3.63). We notice that for $z = \nu_m$ (discrete eigenvalues, $i=1, 2... M$) $A(\nu_i) = 0$ (See equation (3.35)). Consequently, equation (3.72) is well defined if and only if:

$$\int_{-\infty}^{\xi_n} d\xi \frac{\xi \tilde{\psi}(\xi)}{\xi - \nu_i} = 0$$

(3.73)

Introducing the discrete eigenfunctions, according to (3.64), into (3.73) we get:

$$\int_{-\infty}^{\xi_n} d\xi \frac{\xi \tilde{\psi}(\xi)}{\xi - \nu_i} = \int_{-\infty}^{\xi_n} d\xi \frac{\xi}{\xi - \nu_i} \sum_{n=1}^{M} d_m \phi_m(\xi)$$

(3.74)
These M equations \((i = 1, 2, \ldots, M)\) are satisfied if and only if:

\[
d_m = \frac{\int_{\xi_n}^{\xi_{n+1}} \xi d\xi \frac{\partial}{\partial \xi} \phi_{\xi_n}(\xi) \mu(\xi)}{\int_{\xi_n}^{\xi_{n+1}} d\xi [\phi_{\xi_n}(\xi)]^2}
\]

(3.75)

The equations (3.75) are the same values for the expansion coefficients obtained using the full-range orthogonality relationship.

Equations (3.75) complete the proof of the theorem confirming that the eigenfunctions form a complete set for Holder continuous functions.

3.4. Spectral Theory and FN method

The spectral theory outlined in the previous paragraph can now be used to develop the equations that form the basis for the implementation of the FN method. We will show how the orthogonality property of the modes can be used to derive “exact” singular integral equations for the required exit distribution \(I(0, -\xi), \xi > 0\) and \(I(\Delta, \xi), \xi > 0\). The equations can be subsequently solved to compute the desired intensity at the top of the canopy.

We begin by writing the general solution of equation (3.12) in terms of the continuous and discrete modes (For sake of clarity, we omit \(\xi^*\) dependence):

Now, we evaluate the equation (3.76) at \( \tau = 0 \) and \( \tau = \Delta \). Thus, we multiply the equation (3.76) by \( \xi \phi_{\nu, \nu} (\xi) \) (\( \nu \) belongs to the complete spectrum) and we integrate over \([-\xi_m, \xi_m]\). Using the orthogonality property of the modes (3.57) we obtain the following two equations:

\[
\int_{-\xi_m}^{\xi_m} d\xi \xi \phi_{\nu, \nu} (\xi) I(0, \xi) = A(-\nu) N(-\nu)
\]

\[
\int_{-\xi_m}^{\xi_m} d\xi \xi \phi_{\nu, \nu} (\xi) I(\Delta, \xi) = A(-\nu) N(-\nu)e^{\Delta \nu}
\]

In a similar way, we can repeat the procedure using \( \xi \phi_{\nu, \nu} (\xi) \). We obtain:

\[
\int_{-\xi_m}^{\xi_m} d\xi \xi \phi_{\nu, \nu} (\xi) I(0, \xi) = A(\nu) N(\nu)
\]

\[
\int_{-\xi_m}^{\xi_m} d\xi \xi \phi_{\nu, \nu} (\xi) I(\Delta, \xi) = A(\nu) N(\nu)e^{\Delta \nu}
\]
The normalization constants can be eliminated to yield the following singular integral equations:

\[ \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(0, \xi) = e^\frac{\Delta}{\nu} \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(\Delta, \xi) \]  
(3.81)

\[ \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(\Delta, \xi) = e^\frac{\Delta}{\nu} \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(0, \xi) \]  
(3.82)

Using the fact that \( \phi_\nu(\xi) = \phi_\nu(-\xi) \) we can write the more convenient form of the equations (3.81) and (3.82):

\[ \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(0, -\xi) = e^\frac{\Delta}{\nu} \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(\Delta, \xi) = R_1(\xi) \]  
(3.83)

\[ \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(\Delta, \xi) = e^\frac{\Delta}{\nu} \int_{\xi_0}^{\xi} d\xi \xi \phi_\nu(\xi) I(0, -\xi) = R_2(\xi) \]  
(3.84)

Here \( R_1(\xi) \) and \( R_2(\xi) \) are known terms containing the boundary conditions (See (2.9)): 
The two integral equation (3.83)-(3.84) are usually solved by expanding the unknown boundary values in an appropriate basis of functions.

The spectral theory has been developed to provide a theoretical basis for the development of a possible method of solution i.e. the FN method. The FN method has already been implemented to provide a numerical solution for LEAFMOD and CANMOD one-angle formulation [Ref 14, 15]. The next chapter will discuss an alternative scheme implemented to numerically solve both leaf and canopy equations, i.e. the SN method. The former will be compared against the FN method to ensure numerical accuracy and stability of the solution.
4. PART III: NUMERICAL SOLUTION OF THE LEAF AND CANOPY EQUATIONS. DISCRETE ORDINATE METHOD

4.1. Introduction

As seen in section 2, the approach used in modeling the transport of photons in leaf and/or canopy media is based on the application of first principles. For both the leaf and canopy case, the balance of photons within the appropriate phase-space yields the formulation of a local equation describing the physical behavior of the photons in terms of radiant energy. The equations describing the transport for both situations have the same basic structure, i.e., they can be classified as linearized Boltzmann equations. The Boltzmann equation naturally arises in scenarios where transport of particles through a host medium is considered and when the binary collision formulation is assumed appropriate to describe the scattering behavior. The interaction between light and vegetation at the two levels of description yields the same type of equation. The difference lies in the scattering behavior since the transport of the photons in the leaf interior is considered isotropic, while the turbid medium assumption, for the canopy case, defines the transport in canopies as naturally anisotropic. From a mathematical point of view, we are dealing with integro-differential equations. A theoretical analysis consisting in analyzing the spectrum of the canopy transport equation, as shown in section 3, highlighted the extreme complexity and richness of the formulated equation as well as the
mathematical difficulties involved in completing the spectral theory (e.g., proof of completeness).

In this chapter we are interested in performing a numerical investigation of both leaf and canopy equations. The aim is to provide means for computing the radiative regime within and at the top of the canopy. Both canopy and leaf equations cannot be solved in closed form and therefore, if both photon radiance and reflectance, for any defined canopy scenario, are desired, a numerical method of solution must be devised. LEAFMOD and CANMOD are two different but similar models, connected according by the flow chart shown in section 2 (see Figure 9). They are modular in the sense that both models can be independently solved by implementing any appropriate numerical scheme and then connected according to the aforementioned diagram. The method used in providing the solution of the integro-differential equations is an adaptation of the Discrete Ordinate Method (DOM) largely employed in classical neutron transport theory to provide numerical solution of the linearized Boltzmann equation. The discrete ordinate method, also called SN method, has been considered as a possible solution scheme in contrast with an already existing and implemented alternative FN scheme. Ganapol et al. [Ref 14, Ref 15] provided solutions of both leaf and canopy equations in the one-angle formulation using the FN method, which is based on expanding the solution in a basis of orthogonal functions (shifted Legendre polynomials) and, after appropriate truncation, finding the coefficients by solving a linear system of equations. The SN method, as shown in the next sections, is based on discretizing both space and angular variables. The goal is to implement the discrete ordinate method in both LEAFMOD and CANMOD for
one-angle and two-angle formulations. The discrete ordinate equations are formulated and implemented using the MATLAB programming environment. The numerical solutions, for any possible case, are compared to the existing FN numerical solutions which provide the appropriate test-bed for numerical performance analysis.

The following sections describe, in some detail, the SN numerical implementation for both leaf and canopy equations. We start discussing the implementation of the SN method for LEAFMOD one-angle formulation and we extend the discussion to CANMOD one- and two-angle scenario.

4.2. Numerical solution for LEAFMOD: Discrete Ordinate Method for one-angle formulation

In this section we will describe, in some details, the implementation of the Discrete Ordinate Method for the LEAFMOD equation to obtain the numerical evaluation of the hemispherical reflectance and transmittance factors.

The Discrete Ordinate method, also called SN method, is a numerical scheme in which the angular variable is discretized in a small number of angular rays. For each ray, a balance equation is written and every term describing the ray-to-ray transfer is included to capture the essential features of the transport equation. This method is borrowed from conventional neutron transport theory and it is adapted to provide an accurate and suitable numerical solution.

Consider the equation (2.3). The angular variable \( \mu \) is discretized in a set of \( M \) discrete directions (i.e., photons are constrained to travel only in fixed \( M \) directions). For each
direction, a spatial equation can be written yielding the following set of discrete equations
\(m = 1, 2, \ldots, M\):

\[
\left( \mu_m \frac{\partial}{\partial \tau} + 1 \right) I(\tau, \mu_m) = Q(\tau, \mu_m),
\]

(4.1)

where the right hand side of the equation (4.1) is the usual inscattering term

\[
Q(\tau) = \frac{\sigma}{2} \int_{-1}^{1} d\mu' I(\tau, \mu').
\]

(4.2)

The source term \(Q(\tau)\) can be evaluated via quadrature in the following way:

\[
Q(\tau) = \frac{\sigma}{2} \sum_{n=1}^{M} w_n I(\tau, \mu_n)
\]

(4.3)

Here, \(\{w_n\}\) and \(\{\mu_n\}\) are the weights and abscissas of the chosen quadrature scheme.

Indeed, the choice of the particular angular rays is strictly connected to the numerical quadrature used to evaluate the inscattering source. We mean that the abscissas chosen to efficiently compute the angular integral in the transport equation are also forced to be the discrete directions. The domain of integration suggests that the Gauss-Legendre quadrature set is probably the most appropriate for the problem. It is well known that the any Gaussian set integrates exactly a polynomial of a given degree with the least number
of quadrature points. In the case of the Gauss-Legendre, the \( M \)-point scheme integrates exactly a polynomial of order \( 2M-1 \). In addition, the Gauss-Legendre quadrature satisfies the projection invariance and has the positive weights require mandatory for a non-negative integration.

The boundary conditions can be also implemented in terms of discrete directions. The equation (2.3) becomes:

\[
I(0, \mu_n) = \begin{cases} 
(0, m \neq k) \\
\frac{S_0}{w_k}, m = k 
\end{cases} \quad m = 1, 2 \ldots M/2 \tag{4.4}
\]

\[
I(\Delta, \mu_n) = 0 \quad m = M/2 \ldots M \tag{4.5}
\]

The first boundary condition is implemented such that the illuminating source emits photons only at the specified discrete direction. The strength of the source is scaled by the weight associated with the correspondent abscissa to guarantee that the integral over the full range \([-1,1]\) is exactly equal to the source:

\[
\int_{-1}^{1} d\mu I(0, \mu) = \int_{-1}^{1} d\mu S_0 \delta(\mu - \mu_k) = S_0 \Rightarrow \sum_{n=1}^{M} w_n I(0, \mu_n) = w_k I(0, \mu_k) = S_0 \tag{4.6}
\]

Next step is to proceed to the spatial discretization of the equation (4.1). To discretize the spatial variable, we define a spatial grid with \( I \) mesh points as shown in Figure 10. The
\( \tau_1, \tau_2, \ldots, \tau_I \) are the mid-points of the mesh cells. The points are considered equally spaced. The boundaries of the \( i^{th} \) cell are labeled as \( \tau_{i+1/2} \). Now, the spatial discretization is implemented in the equation (4.1). For each mesh cell, the derivative is approximated as follows:

\[
\frac{\partial I}{\partial \tau} \approx \frac{I(\tau_{i+1/2}, \mu) - I(\tau_{i-1/2}, \mu)}{\Delta \tau} \quad (4.7)
\]

Figure 10: Leaf spatial discretization scheme.
Inserting (4.7) into equation (4.1) we have:

\[
\mu_m \left( \frac{I\left(\tau_{i+\frac{1}{2}}, \mu_m\right) - I\left(\tau_{i-\frac{1}{2}}, \mu_m\right)}{\Delta \tau} \right) + I(\tau_i, \mu_m) = \frac{\omega}{2} \sum_{n=1}^{M} w_n I(\tau_i, \mu_n),
\]

(4.8)

where the notation can be simplified yielding the following equation:

\[
\mu_m \left( \frac{I_{m+\frac{1}{2}}^{i-1} - I_{m-\frac{1}{2}}^{i-1}}{\Delta \tau} \right) + I_m^i = \frac{\omega}{2} \sum_{n=1}^{M} w_n I_n^i = Q_m^i.
\]

(4.9)

The above expressions represent a set of \(1^*M\) equations in \((2^*1 + 1)^*M\) unknown because (4.9) includes the cell-centered fluxes as well as the cell-edge fluxes. To close the systems, we need auxiliary relationships to relate the cell-centered fluxes with the cell-edged fluxes. We choose the so-called "Diamond Difference" relations:

\[
I_m^i = \frac{1}{2} \left( I_{m+\frac{1}{2}}^{i-1} + I_{m-\frac{1}{2}}^{i-1} \right)
\]

(4.10)
Equations (4.9) and (4.10) can be combined to determine the final form of the difference equations which allow the numerical computation of the interior radiant intensity as well as the exiting radiant flux. We distinguish two cases depending on $\mu_m > 0$ or $\mu_m < 0$.

If $\mu_m > 0$ we eliminate $I_m^{/2}$ from (4.9) to get:

$$I_i^{i-1} = \left(1 + \frac{\Delta r}{2|\mu_m|}\right)^{-1} \left(I_m^{i-1} - \frac{\Delta r}{2|\mu_m|} Q_m^i\right)$$

(4.11)

and

$$I_m^{i-1} - 2I_m^i + I_m^{i+1} = 0$$

(4.12)

If $\mu_m < 0$ we eliminate $I_m^{/2}$ from (4.9) to get:

$$I_i^{i-1} = \left(1 + \frac{\Delta r}{2|\mu_m|}\right)^{-1} \left(I_m^{i-1} + \frac{\Delta r}{2|\mu_m|} Q_m^i\right)$$

(4.13)

and

$$I_m^{i-1} - 2I_m^i - I_m^{i+1} = 0.$$  

(4.14)

These equations can be solved by standard inward-outward sweep techniques. We start with the boundary value of the intensity at the left-hand side of the leaf and move in the inward direction ($\mu_i > 0$). For each $\mu_i > 0$ we march rightward in space until we hit the
right boundary. Then, for each $\mu_i < 0$ we march in space leftward until we hit the left boundary. For each step the inscattering source is updated. This procedure represents just one iterate in which the full domain has been swept both in space and inclination. At the end of each iterate, the exiting reflectance is computed by implementing the Gauss-Legendre quadrature to solve the equation (2.5). The inward-outward iteration is continued until a convergence criterion is satisfied. The stopping condition is based on the evaluation on the Euclidean and infinite norm, i.e., the norm of the difference between two hemispherical reflectance are computed for two consecutive iterations and the procedure arrested when one of the errors is less than $10^{-10}$.

4.2.1. Numerical implementation via zero weights concept: Comparison with the FN solution

The method, as described above, was coded and implemented in a MATLAB script. The goal was to provide a numerical solution for the reflectance and transmittance. A baseline maple leaf is chosen and the SN method employed to compute the leaf optical properties. The SN solution is compared with the solution obtained employing the FN method. One technical issue faced whenever dealing with discretizing the angular variable is the problem of providing the boundary conditions at a very specific angle that is not part of the chosen discrete set. In our case, the left boundary conditions describe a source illuminating the boundary at a certain angle $\mu_0$. For the sake of numerical simulation, the illuminating angle was chosen to be zero. The set of discrete angles is dictated by the numerical quadrature chosen to compute the inscattering source. If Gauss-
Legendre quadrature set of order M (i.e. M angular directions) is chosen, no discrete direction in the set will be zero no matter how large the M value. The larger the order of quadrature, the more the directions and the closer the smallest angle in the set will be closer to zero.

To tract the aforementioned consideration, we implemented the so called “zero weight technique”. In essence the Gaussian set will be incremented in two additional angular directions, one corresponding to the point we intend to implement the mono-directional boundary condition source and the negative symmetric direction cosine (the negative symmetric is required to maintain symmetry in the numerical scheme). The essence of the method lies in the choice of the weights associated with the additional discrete angles: the weights will be set to zero so that their introduction will not affect the overall computation but will just be a way to edit boundary conditions (indeed, the weights value was set equal to $10^{-10}$ to avoid numerical problems).

To provide a test bed for the SN numerical implementation, we considered the specific task of reconstructing the optical properties of the maple leaf (reflectance and transmittance) from the given scattering and absorption profile. The same computation was performed by solving the LEAFMOD one angle equation via FN method [Ganapol et al]. The leaf under consideration is a maple leaf with thickness of 0.0134 cm and the scattering profile was specified via the LEAFMOD inversion mode using the reference maple leaf optical properties coming from the LOPEX archive. The absorption profile was constructed by specifying the concentration of chlorophyll, water, cellulose and lignin inside the leaf. The scattering and absorption coefficients were provided in the
wavelength range 400-790 nm. The computation was performed for 40 wavelengths inside the above range (wavelengths equally spaced with 10 nm interval). Figure 11 and Figure 12 show the reflectance and transmittance as function of the wavelengths with discrete ordinate order as parameter. The idea is to test the accuracy of the results as well as the effect of the angular discretization (i.e., what is the effect of increasing the number of angles in the discrete set?). There are 5 lines in both plots: the blue line is the spectral profile computed using the FN method, while the other lines are the profiles computed using SN with increasing numbers of angles (8, 16, 32, and 64). The lines are practically undistinguishable because the error is extremely small. Figure 13 and Figure 14 show the detailed behavior of the absolute error between FN, assumed to be the baseline solution, and SN method with different integration orders. We notice that the error is less than $10^{-4}$ in any case. In general, the error tends to be smaller with increasing M. Nevertheless, for M=64 the error is higher. It is good to be reminded that the baseline for comparison is the FN method which is a numerical procedure as well, and therefore we do not expect the error to reduce to zero as M increases because we are not comparing with a closed form analytical solution.

The procedure can be repeated to analyze the behavior of the numerical solution as a function of the spatial discretization. Figure 15 and Figure 16 show the reflectance and transmittance as function of wavelength with the number of mesh-points (I), chosen to discretize the leaf domain, as parameter. The mesh-grid numbers I assume four possible values (32, 64, 128, 256). Figure 17 and Figure 18 show the absolute error behavior. The latter tends to zero as the number of mesh-points is increased.
Comparing SN with FN: Leaf Reflectance vs wavelength

Figure 11: LEAFMOD FN and SN computed Reflectance at various wavelengths and spatial discretization effect.

Comparing SN with FN: Leaf Transmittance vs wavelength

Figure 12: LEAFMOD FN and SN computed Transmittance at various wavelengths and spatial discretization effect.
Comparing SN with FN: Reflectance Error vs wavelength

Figure 13: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of spatial points as parameter).

Comparing SN with FN: Transmittance Error vs wavelength

Figure 14: Comparing FN and SN results. Transmittance absolute error at various wavelengths (number of spatial points as parameter).
Figure 15: LEAFMOD FN and SN computed Reflectance at various wavelengths and angular discretization effect.

Figure 16: LEAFMOD FN and SN computed Transmittance at various wavelengths and angular discretization effect.
Figure 17: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of angular rays as parameter).

Figure 18: Comparing FN and SN results. Transmittance absolute error at various wavelengths (number of angular rays as parameter).
4.3. Discrete Ordinate Method for CANMOD: One-angle formulation

The canopy equation can also be solved using the discrete ordinate method. The methodology used in devising the CANMOD discrete equations is conceptually similar to the LEAFMOD case, although special attention must be paid to the way area scattering function $\Gamma(\mu, \mu')$ is implemented. This section illustrates the steps required for a proper numerical implementation of CANMOD one-angle SN method.

Consider equation (2.15) and the associated boundary conditions (2.16). As in the LEAFMOD implementation, the equation must be discretized both in the angular variable and in space. Consider $M$ discrete directions for the inclination $\mu$. We start one equation for each angular ray ($m = 1, 2, \ldots, M$):

\[
\left[ \mu_m \frac{\partial}{\partial \tau} + G(\mu_m) \right] I(\tau, \mu_m) = Q(\tau, \mu_m) \quad (4.15)
\]

Where

\[
Q(\tau, \mu_m) = \int_{-1}^{1} d\mu' \Gamma(\mu', \mu_m) I(\tau, \mu') \quad (4.16)
\]

The inscattering term $Q(\tau, \mu_m)$ contains an integral which must be numerically solved. The limits of integration suggest the use of the Gauss-Legendre quadrature scheme as in the LEAFMOD case. The scheme yields the following discrete equations:
Here \( \{w_n\} \) and \( \{\mu_n\} \) are the weights and abscissas of the chosen Gauss-Legendre quadrature scheme. The area scattering function \( \Gamma(\mu', \mu) \) is written for \( 2^n M \) discrete directions in the two angular variables \( (n, m = 1, 2, \ldots, M) \):

\[
Q(\tau, \mu_m) = \sum_{n=1}^{M} w_n \Gamma(\mu_n, \mu_m) l(\tau, \mu_n)
\]  
(4.17)

The limits of integration permit the use of the conventional Gauss-Legendre quadrature scheme. We have \( (k = 1, 2, \ldots, M_g) \):

\[
\Gamma(\mu_n, \mu_m) = \int_{-1}^{1} d\mu_t g_L(\mu_t) \alpha(\mu_n', \mu_l) b(\mu_m, \mu_l)
\]  
(4.18)

The limits of integration permit the use of the conventional Gauss-Legendre quadrature scheme. We have \( (k = 1, 2, \ldots, M_g) \):

\[
\Gamma(\mu_n, \mu_m) = \sum_{k=1}^{M_g} g_L(\mu_k) \alpha(\mu_n', \mu_k) b(\mu_m, \mu_k)
\]  
(4.19)

The \( M_g \) index is the chosen order of quadrature of the integration in the normal leaf inclination. Numerical experiments show that the number of quadrature points in this variable must be higher than the quadrature points in the photons angular variable to obtain a good accuracy.
The numerical integration of the scattering area function involves the use of the function $H(\mu, \mu_L)$ (See (2.21) and (2.22)). This function appears frequently in photons transport within vegetation whenever the leaf is assumed a bi-Lambertian scattering center. Figure 19 shows the contour plot of the function derived from our numerical implementation which is identical to the function plotted by Shultis [Ref 31].

![Contour plot](image)

Figure 19: $H(\mu, \mu_L)$ contour plot

The intercept function $G(\mu_m)$ must be obtained through numerical quadrature:

$$G(\mu_m) = \int d\mu_L g_L(\mu_L) \nu(\mu_m, \mu_L)$$

(4.20)
The function must be integrated over the normal leaf inclination range [0, 1]. Since the limits of integration are not -1 and 1, the conventional Gauss-Legendre quadrature scheme cannot be applied. One possible way to approach this integration is to perform a linear transformation that maps the [-1, 1] range into the [0, 1]. The idea is to use the same weights for abscissas mapped into the range [0, 1]. This quadrature scheme is commonly known as Shifted Gauss-Legendre quadrature.

The boundary conditions must be implemented using the discrete directions as well:

\[ I(0, \mu_m) = F(\mu_m) \quad \text{For } m = 1, 2, \ldots, M/2 \]  \hspace{1cm} (4.21)

\[ I(\Delta, \mu_m) = 2r \sum_{k=1}^{M/2} w_k \mu_k I(\Delta, \mu_k) \quad \text{For } m = M/2, \ldots, M \]  \hspace{1cm} (4.22)

The bottom boundary condition is defined by (2.16). It corresponds to imposing an isotropic boundary reflection. The integral in the range [-1, 0] is integrated by using the Double Gauss-Legendre quadrature weights and abscissas.

Spatial discretization is the next step. The canopy is divided into I layers which defines the spatial grid. Figure 20 shows a graphic representation of how the spatial discretization is performed.
Figure 20: Canopy spatial discretization scheme.

The steps required to devise the final discrete equations including the spatial contribution are identical to the steps taken in the LEAFMOD case (See equation (4.7) to (4.10)) and will not be repeated. The final discrete equations for the CANMOD one-angle formulations are the following:

Case $\mu > 0$ ($i = 1, 2, \ldots, I; m = 1, 2, \ldots, M/2$):

\[
I_m^{i+1} = \left(1 + \frac{G(\mu_m) \Delta \tau}{2|\mu_m|} \right)^{-1} \left( I_m^{i-\frac{1}{2}} + \frac{\Delta \tau}{2|\mu_m|} Q_m' \right)
\]  

(4.23)

\[
I_m^{i+\frac{1}{2}} = 2I_m^{i} - I_m^{i-\frac{1}{2}}
\]

Case $\mu < 0$ ($i = 1, 2, \ldots, I; m = M/2, \ldots, M$)
These equations can be solved by standard inward-outward sweep techniques as with the LEAFMOD case. We start with the boundary condition of the intensity at the top of the canopy, which represents the illumination source from direct sunlight and diffusive sky light, and move in the inward (downward) direction ($\mu_m > 0$). For each $\mu_m > 0$ we march in space until we hit the bottom boundary (soil). For each $\mu_m < 0$ we march in space upward until we hit the top of the canopy. The in-scattering source is updated at each step. This procedure represents just one iterate in which the full domain has been swept both in space and inclination angle. At the end of each iterate, the exiting reflectance is computed by implementing the Shifted Gauss-Legendre quadrature to compute the reflectance:

$$R_f = \frac{1}{\mu_0} \int_{\mu_0}^{1} d\mu' \mu' I(0, -\mu') \approx \sum_{k=1}^{M} w_k \mu_k I(0, \mu_k)$$

(4.25)
norms of the difference between two hemispherical reflectances are computed for two consecutive iterations and the procedure arrested when one of the norms is less than $10^{-10}$.

4.3.1. Numerical implementation: Comparison CANMOD SN versus FN solution

The discrete ordinate equations (4.23)-(4.24), derived by discretizing both spatial and angular variables, are coded and implemented on a digital computer to compute, for any desired wavelength, the hemispherical reflectance at the top of the canopy. As in the LEAFMOD case, we used the MATLAB programming environment to provide the appropriate solution. The numerical solution obtained using the SN method must be tested to verify the reliability of the methodology. The Boltzmann equation, defining CANMOD in its one-angle formulation, was previously solved by using the FN method [Ganapol Ref 15]. The FN method provides an excellent test for the SN reflectance computation.

We start by defining a canopy scenario. We considered a maple canopy with normal leaves mostly pointing in the zenith (planophile). We considered 10 wavelengths (400-850 nm) and we used LEAFMOD to determine the leaf reflectance and transmittance (Ref and Trs). Table 2 shows the CANMOD input data for both FN and SN simulations which include LAI, sun angle (SA), soil reflectance (SR) as well. For each wavelength the reflectance TOC is computed using both methods and the results compared. Moreover the SN simulation is performed varying both angular and spatial mesh grids. Three possible angular and spatial sets are chosen. For the angular variable, we consider 16, 32 and 64 possible discrete directions while for the spatial variable the three grids
considered contained 20, 60 and 120 points. The performed SN multiple computations will help to analyze the behavior of the solution. Figure 23 and Table 3 show the absolute error behavior as function of the angular discretization. The worst case occurs in the near-infrared (wavelength 800 and 850 nm) and using 16 angular rays where the absolute error is $10^{-3}$. In the other cases, the errors vary in the range $10^{-4} - 10^{-6}$. Ideally, we would like to see the absolute error going to zero as the number of rays increases. This is not possible since the SN computation is not compared with the exact solution but with the FN numerical solution. The latter is affected by numerical errors such as truncation and round-off. Nevertheless, the two methods are very close in the range of the considered wavelengths. Figure 24 also reports the relative error behavior.

<table>
<thead>
<tr>
<th>Lambda</th>
<th>Leaf Ref</th>
<th>Leaf Trs</th>
<th>LAD</th>
<th>LAI</th>
<th>SR</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>0.0417</td>
<td>0.0010</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>450</td>
<td>0.0486</td>
<td>0.0111</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>500</td>
<td>0.0568</td>
<td>0.0313</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>550</td>
<td>0.1430</td>
<td>0.1862</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>600</td>
<td>0.0843</td>
<td>0.1160</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>650</td>
<td>0.0609</td>
<td>0.0509</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>700</td>
<td>0.1413</td>
<td>0.1657</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>750</td>
<td>0.5030</td>
<td>0.4270</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>800</td>
<td>0.5151</td>
<td>0.4394</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>zenith</td>
</tr>
<tr>
<td>850</td>
<td>0.5174</td>
<td>0.4435</td>
<td>Planophile</td>
<td>1</td>
<td>0.3</td>
<td>Zenith</td>
</tr>
</tbody>
</table>

Table 2: CANMOD input data for maple-canopy simulation scenes

The same analysis is performed by increasing the number of grid-points. Figure 25 and Table 4 report the absolute error as function of the wavelength and with mesh-points
number as parameter. The absolute error is never higher, for any considered spatial-
dependent situation, than \(10^{-4}\). Figure 26 shows the associated relative error.

Computational time is also reported. To provide both SN and -N solution, for any
possible case, we used a PC equipped with a dual processor (Athlon 1.2 GHz each) and
724 MB RAM. The FN method was written using Fortran77 and it takes approximately
0.25 seconds of CPU time for each case. The MATLAB simulation CPU time, as
reported in Table 5, is much longer and ranges from 14 sec using 16 angular rays to 224
sec using 64 angular rays. Indeed, it is very well known that MATLAB runs slower than
Fortran77 although programming and implementations are easier to perform.
Comparing SN with FN: Reflectance vs wavelength; Angular Discretization

**Figure 21:** CANMOD FN and SN computed Reflectance at various wavelengths and angular discretization effect.

Comparing SN with FN: Reflectance vs wavelength; Spatial Discretization

**Figure 22:** CANMOD FN and SN computed Reflectance at various wavelengths and spatial discretization effect.
Figure 23: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of angular rays as parameter).

Figure 24: Comparing FN and SN results. Reflectance relative error at various wavelengths (number of angular rays as parameter).
Comparing SN with FN: Spatial discretization effect.

Figure 25: Comparing FN and SN results. Reflectance absolute error at various wavelengths (number of spatial points as parameter).

Figure 26: Comparing FN and SN results. Reflectance relative error at various wavelengths (number of spatial points as parameter).
Comparing F-N and S-N method

<table>
<thead>
<tr>
<th>Lambda</th>
<th>Ref FN</th>
<th>Ref S-N M=16</th>
<th>Ref S-N M=32</th>
<th>Ref S-N M=64</th>
<th>Error M=16</th>
<th>Error M=32</th>
<th>Error M=64</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>0.0656</td>
<td>0.0650</td>
<td>0.0656</td>
<td>0.0658</td>
<td>5.841e-4</td>
<td>9.978e-6</td>
<td>2.387e-4</td>
</tr>
<tr>
<td>450</td>
<td>0.0693</td>
<td>0.0687</td>
<td>0.0693</td>
<td>0.0695</td>
<td>5.291e-4</td>
<td>2.742e-5</td>
<td>2.434e-4</td>
</tr>
<tr>
<td>500</td>
<td>0.0746</td>
<td>0.0742</td>
<td>0.0747</td>
<td>0.0749</td>
<td>4.326e-4</td>
<td>5.780e-5</td>
<td>4.503e-4</td>
</tr>
<tr>
<td>550</td>
<td>0.1330</td>
<td>0.1334</td>
<td>0.1333</td>
<td>0.1333</td>
<td>4.084e-4</td>
<td>3.102e-4</td>
<td>2.977e-4</td>
</tr>
<tr>
<td>600</td>
<td>0.0964</td>
<td>0.0964</td>
<td>0.0967</td>
<td>0.0968</td>
<td>1.080e-5</td>
<td>2.247e-4</td>
<td>3.186e-4</td>
</tr>
<tr>
<td>650</td>
<td>0.0785</td>
<td>0.0782</td>
<td>0.0786</td>
<td>0.0788</td>
<td>3.326e-4</td>
<td>9.920e-5</td>
<td>2.704e-4</td>
</tr>
<tr>
<td>700</td>
<td>0.1286</td>
<td>0.1289</td>
<td>0.1288</td>
<td>0.1289</td>
<td>2.971e-4</td>
<td>2.617e-4</td>
<td>2.725e-4</td>
</tr>
<tr>
<td>750</td>
<td>0.4166</td>
<td>0.4180</td>
<td>0.4164</td>
<td>0.4158</td>
<td>1.374e-3</td>
<td>1.926e-4</td>
<td>7.910e-4</td>
</tr>
<tr>
<td>800</td>
<td>0.4327</td>
<td>0.4341</td>
<td>0.4324</td>
<td>0.4318</td>
<td>1.363e-3</td>
<td>2.538e-4</td>
<td>8.819e-4</td>
</tr>
<tr>
<td>850</td>
<td>0.4368</td>
<td>0.4382</td>
<td>0.4365</td>
<td>0.4359</td>
<td>1.353e-3</td>
<td>2.774e-4</td>
<td>9.136e-4</td>
</tr>
</tbody>
</table>

Table 3: Comparing FN and SN. Results and absolute errors for various angular discretization points.

Comparing F-N and S-N method

<table>
<thead>
<tr>
<th>Lambda</th>
<th>Ref FN</th>
<th>Ref S-N I=20</th>
<th>Ref S-N I=60</th>
<th>Ref S-N I=120</th>
<th>Error I=20</th>
<th>Error I=60</th>
<th>Error I=120</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>0.0656</td>
<td>0.0654</td>
<td>0.0654</td>
<td>0.0654</td>
<td>1.973e-4</td>
<td>1.842e-4</td>
<td>1.830e-4</td>
</tr>
<tr>
<td>450</td>
<td>0.0693</td>
<td>0.0691</td>
<td>0.0691</td>
<td>0.0691</td>
<td>1.679e-4</td>
<td>1.551e-4</td>
<td>1.539e-4</td>
</tr>
<tr>
<td>500</td>
<td>0.0746</td>
<td>0.0745</td>
<td>0.0745</td>
<td>0.0745</td>
<td>1.160e-4</td>
<td>1.038e-4</td>
<td>1.026e-4</td>
</tr>
<tr>
<td>550</td>
<td>0.1330</td>
<td>0.1333</td>
<td>0.1333</td>
<td>0.1333</td>
<td>3.259e-4</td>
<td>3.335e-4</td>
<td>3.342e-4</td>
</tr>
<tr>
<td>600</td>
<td>0.0964</td>
<td>0.0966</td>
<td>0.0966</td>
<td>0.0966</td>
<td>1.409e-4</td>
<td>1.512e-4</td>
<td>1.520e-4</td>
</tr>
<tr>
<td>650</td>
<td>0.0785</td>
<td>0.0785</td>
<td>0.0785</td>
<td>0.0785</td>
<td>5.541e-5</td>
<td>4.356e-5</td>
<td>4.245e-5</td>
</tr>
<tr>
<td>700</td>
<td>0.1286</td>
<td>0.1288</td>
<td>0.1288</td>
<td>0.1288</td>
<td>2.566e-4</td>
<td>2.647e-4</td>
<td>2.654e-4</td>
</tr>
<tr>
<td>750</td>
<td>0.4166</td>
<td>0.4169</td>
<td>0.4169</td>
<td>0.4169</td>
<td>3.194e-4</td>
<td>3.192e-4</td>
<td>3.192e-4</td>
</tr>
<tr>
<td>800</td>
<td>0.4327</td>
<td>0.4330</td>
<td>0.4330</td>
<td>0.4330</td>
<td>2.782e-4</td>
<td>2.784e-4</td>
<td>2.784e-4</td>
</tr>
<tr>
<td>850</td>
<td>0.4368</td>
<td>0.4371</td>
<td>0.4371</td>
<td>0.4371</td>
<td>2.601e-4</td>
<td>2.604e-4</td>
<td>2.604e-4</td>
</tr>
</tbody>
</table>

Table 4: Comparing FN and SN. Results and absolute errors for various spatial discretization points.
### Computational time for the S-N MALAB implementation

<table>
<thead>
<tr>
<th>Lambda</th>
<th>Angular Discretization</th>
<th>Spatial Discretization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (sec)</td>
<td>Time (sec)</td>
</tr>
<tr>
<td>M=16</td>
<td></td>
<td>M=32</td>
</tr>
<tr>
<td>400</td>
<td>14.75</td>
<td>75.62</td>
</tr>
<tr>
<td>450</td>
<td>14.76</td>
<td>75.80</td>
</tr>
<tr>
<td>500</td>
<td>14.84</td>
<td>75.94</td>
</tr>
<tr>
<td>550</td>
<td>15.69</td>
<td>77.64</td>
</tr>
<tr>
<td>600</td>
<td>15.23</td>
<td>76.86</td>
</tr>
<tr>
<td>650</td>
<td>15.07</td>
<td>76.21</td>
</tr>
<tr>
<td>700</td>
<td>15.53</td>
<td>77.24</td>
</tr>
<tr>
<td>750</td>
<td>21.47</td>
<td>82.78</td>
</tr>
<tr>
<td>800</td>
<td>23.27</td>
<td>82.67</td>
</tr>
<tr>
<td>850</td>
<td>23.15</td>
<td>82.63</td>
</tr>
</tbody>
</table>

Table 5: MATLAB SN implementation: CPU time.
4.4. Discrete Ordinate Method for CANMOD: Two-angle formulation

The two-angle formulation for the canopy model has been described in section 2. The balance of photons yielded equation (2.8) which must be discretized to compute the reflectance at Top-Of-the-Canopy (TOC) and the radiative regime within the canopy structure. The radiant intensity, which is the desired unknown, now depends on three variables, i.e., one space coordinate \( \tau \) and two parameters to describe the direction unit vector \( \bar{\Omega} = (\mu, \varphi) \). The discrete ordinate method can be applied in a straightforward manner also in this situation. The steps taken in deriving the discrete equations are identical to the one-angle formulation case. Figure 28 shows the unit sphere. In the two angle formulation, there are an infinite possible photons directions in the \( 4\pi \) solid angular range. The first step is to consider a set of discrete directions \( \bar{\Omega}_j \) and write one equation for each direction:

\[
\mu \frac{\partial}{\partial \tau} I(\tau, \bar{\Omega}_j) + G(\bar{\Omega}_j) I(\tau, \bar{\Omega}_j) = Q(\tau, \bar{\Omega}_j) \tag{4.26}
\]

Here, \( \bar{\Omega}_j = (\mu_j, \varphi_j) \) are chosen by independently discretizing the inclination and the azimuth, i.e., \( N \) points are chosen for the inclination in the range \([-1, 1]\) and \( M \) points are chosen for the azimuth varying in the range \([0, 2\pi]\). The total number of angular rays is therefore \( N \times M \). The right hand side represents the in-scattering term and can be written as follows:
A double quadrature scheme is required to numerically evaluate this double integral. The limits of integration in $\mu$ suggest the use of Gauss-Legendre quadrature scheme choosing the appropriate weights and abscissas according to the scheme followed in the previous sections. The limits of integration in the azimuthal variable give us different possible choices in choosing the appropriate scheme. We decided to use the Gauss-Chebyshev quadrature to perform the azimuthal integration. In general terms, the scheme can be summarized as follow:

\[
\int_{a}^{b} f(x) \, dx = \int_{-1}^{1} \frac{g(\xi)}{\sqrt{1-\xi^2}} \, d\xi = \sum_{k=1}^{M} w(\xi_k) g(\xi_k) + R_M(\xi)
\]  

(4.28)

\[
g(\xi) = \frac{b-a}{2} \sqrt{1-\xi^2} \int \left( \frac{b-a}{2} \xi + \frac{b+a}{2} \right)
\]  

(4.29)

The integration of a function $f(x)$ between $a$ and $b$ can be obtained by transforming the problem into an integration between $-1$ and $1$ according to the formulas (4.28) and (4.29). In line with this scheme, the weights depend on the order of quadrature and can be written as:
The abscissas are the zeros of the Chebyshev polynomials of order $M$:

$$\xi_k = \cos\left(\frac{(2k - 1)\pi}{M}\right)$$  \hspace{1cm} (4.31)

With this double-quadrature scheme in place, the in-scattering term can be written as:

$$Q(\tau, \Omega_y) = \frac{1}{\pi} \sum_{m=1}^{M} w_m \sum_{n=1}^{N} \bar{w}_n \Gamma(\Omega_{m\tau}, \Omega_y) I(\tau, \Omega_y)$$  \hspace{1cm} (4.32)

Here $w_m$ are the Gauss-Chebyshev weights while $\bar{w}_n$ are the Gauss-Legendre weights.

The area scattering function must now be evaluated. Indeed, for any coupled of discrete directions, another double integral must be solved. According to the definition, we can write the area scattering function as:

$$\frac{1}{\pi} \Gamma(\Omega', \Omega) = \int_0^1 d\mu_L \int_0^{2\pi} d\phi_L \mid \Omega \cdot \Omega_L \mid |g_L(\Omega_L)| r(\Omega', \Omega; \Omega_L)$$  \hspace{1cm} (4.33)

The integration is performed over the range of the normal leaf. The limits of integration in the $\mu_L$ variable suggest the implementation of the shifted Gauss-Legendre scheme,
while the integration in the azimuth is the Gauss-Chebyshev scheme. With this scheme, we can write

\[
\frac{1}{\pi} \Gamma(\Omega_{xy}, \Omega_{y}) = \sum_{\ell=1}^{N} w_{\ell} \sum_{m=1}^{M} \overline{w}_{m} \left| \Omega_{y} \cdot \Omega_{rs} \right| g_{\ell} (\Omega_{rs}) \phi \left( \Omega_{nm}, \Omega_{y} ; \Omega_{rs} \right)
\]

Here the weights \( w_{\ell}, \overline{w}_{m} \) are the weights for the shifted Gauss-Legendre and Gauss-Chebyshev respectively. Figure 28 shows the contour plot of the scattering area function. In the plot, the direction \( \overline{\Omega} \) is fixed and the scattering area function varies with \( \overline{\Omega} \) only.

As seen in section 2, the intercept function depends only on the inclination angle due to the assumption of random leaf distribution in the azimuthal angle. \( G(\mu) \) is computed in the same way we did for the one-angle formulation case.

The boundary conditions must be implemented using the discrete directions as well:

\[
I(0, \Omega_{y}) = F(\Omega_{y})
\]

\[
I(\Delta, \Omega_{y}) = \frac{P_{s}}{\pi} \sum_{m=1}^{N} W_{m} \sum_{n=1}^{M} \overline{W}_{m} \mu_{n} I(\Delta, \Omega_{nm})
\]

The bottom boundary condition is defined by (2.9). It corresponds to imposing an isotropic boundary reflection.
Figure 27: Unit sphere. The photon direction is described by azimuth and inclination.

Figure 28: Area scattering function contour plot.
The integral in the [-1, 0] range is computed by using the Shifted Gauss-Legendre quadrature weights and abscissas while the integration in the azimuthal variable is performed using the above mentioned Gauss-Chebyshev scheme.

The steps required to devise the final discrete equations including the spatial contribution are identical to the steps taken in the CANMOD and LEAFMOD cases (See equation (4.7) to (4.11)) and will not be repeated. The final discrete equations for the CANMOD two-angle formulations are written as follows:

**Case \( \mu > 0 \):**

\[
I^i_{n,m} = \left(1 + \frac{G(\mu_n)\Delta \tau}{2|\mu_n|}\right)^{-1}\left(I^{i+\frac{1}{2}}_{n,m} + \frac{\Delta \tau}{2|\mu_n|}Q'_{n,m}\right) \tag{4.36}
\]

\[
I^{i+\frac{1}{2}}_{n,m} = 2I^i_{n,m} - I^{i-\frac{1}{2}}_{n,m}
\]

**Case \( \mu < 0 \):**

\[
I^i_{n,m} = \left(1 + \frac{G(\mu_n)\Delta \tau}{2|\mu_n|}\right)^{-1}\left(I^{i+\frac{1}{2}}_{n,m} + \frac{\Delta \tau}{2|\mu_n|}Q'_{n,m}\right) \tag{4.37}
\]

\[
I^{i+\frac{1}{2}}_{n,m} = 2I^i_{n,m} - I^{i-\frac{1}{2}}_{n,m}
\]
Note that we have slightly changed indices. In the above equations we set
\[ I(\tau, \Omega_{\text{row}}) = I'_{n,m} \quad \text{and} \quad Q(\tau, \Omega_{\text{row}}) = Q'_{n,m}. \]

Equations (4.37) and (4.38) are the discrete equations for CANMOD two-angle formulation. They are solved according to the conventional "sweeping technique".

Figure 29 illustrates the three phases required for correct implementation. We start from the top of the canopy where the illuminating boundary conditions are available. For each \( \mu > 0 \), the radiant intensity at the next spatial point (center-cell intensity and edge-cell intensity) is computed for all \( M \) possible azimuthal angles using (4.37). The procedure is repeated marching in space (See B in Figure 29). When the bottom is reached, the soil boundary conditions are applied to compute the radiant intensity for \( \mu < 0 \) at the bottom.

This boundary condition is used to compute the radiant intensity at upper spatial levels.

In this situation, at any level, for any \( \mu < 0 \), we compute the intensity for all azimuthal discrete angles at the next upper spatial point using equations (4.38). When the top of the canopy is reached, the hemispherical reflectance is computed. The letter marks the end of the iteration. The computed reflectance at the \( n^{\text{th}} \) iteration is compared with the previous \( (n-1)^{\text{th}} \) computed reflectance. The procedure is stopped when the absolute difference between two iterates is less than \( 10^{-5} \) (stopping tolerance).
Figure 29: Sweeping technique sequence of operations.
4.4.1. Numerical implementation: Two-angles versus One-angle

The discrete equations derived by discretizing the equation defining the CANMOD two-angle formulation were coded and implemented using the MATLAB programming environment. The goal of this section is to present the numerical results obtained in computing, for given wavelengths, both radiant intensity and hemispherical reflectance and compare the latter with data coming from the one-angle formulation which has already been tested in the section 4.3.1.

Table 6 shows the parameters used in setting four possible maple canopy scenarios. We considered four wavelengths (450, 550, 650 and 800 nm) and for each wavelengths we ran LEAFMOD module to compute the maple leaf optical properties (reflectance and transmittance). The LAI and LAD (canopy architecture) and soil reflectance were set as in the table. The sun angle was defined assigning both inclination and azimuth. These parameters are forwarded to the two-angle module as input variables. We defined both angular and spatial grid by assigning 10 values for both azimuth and inclination and 60 spatial cells. The angular abscissas were chosen according to the Gauss-Legendre and Gauss-Chebyshev prescription. The independent angular discretization in both inclination and azimuth yields a total of 100 discrete rays. The CANMOD two-angle algorithm is run to compute the hemispherical reflectance at the four defined wavelengths. The hemispherical reflectance in compared with the results obtained by running the azimuthally-averaged one-angle formulation and the results compared. Table 7 and Table 8 show the results obtained in both one-angle and two-angle cases.
Table 6: Canopy input data for one-angle and two angle CANMOD simulation.

<table>
<thead>
<tr>
<th>Wave</th>
<th>Leaf Ref</th>
<th>Leaf Trs</th>
<th>Soil Ref</th>
<th>Sun Inclin.</th>
<th>Sun Azimuth</th>
<th>LAI</th>
<th>LAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>4.8603E-2</td>
<td>1.1097E-2</td>
<td>0.3</td>
<td>13</td>
<td>98</td>
<td>1</td>
<td>Planophile</td>
</tr>
<tr>
<td>550</td>
<td>1.4306E-1</td>
<td>1.8621E-1</td>
<td>0.3</td>
<td>13</td>
<td>98</td>
<td>1</td>
<td>Planophile</td>
</tr>
<tr>
<td>650</td>
<td>6.0893E-2</td>
<td>5.0899E-2</td>
<td>0.3</td>
<td>13</td>
<td>98</td>
<td>1</td>
<td>Planophile</td>
</tr>
<tr>
<td>800</td>
<td>5.1512E-1</td>
<td>4.3943E-1</td>
<td>0.3</td>
<td>13</td>
<td>98</td>
<td>1</td>
<td>Planophile</td>
</tr>
</tbody>
</table>

For each canopy case, the wavelength, the number of angular rays, the convergence tolerance and the correspondent computed hemispherical reflectance are reported. Moreover, we reported the number of iterates required to reach convergence as well as the CPU computational time. The results are comparable. Indeed, the reflectance values should be comparable since the hemispherical reflectance is integrated over the upper hemisphere in both cases. Table 9 shows the error analysis results. We see that the two angle formulation compute the same reflectance value with an absolute error that ranges between $10^{-3}$ and $10^{-4}$. These results are a confirmation of the correct and reliable implementation. For further error analysis, we decided to consider the last wavelength (which is the worst case scenario) and we held the inclination discretization to 10 values, increasing the number of azimuthal points to 12, 14 and 16 respectively. For each of these cases, we computed the hemispherical reflectance and compared it with the results obtained from the one-angle algorithm using the same canopy inputs and 10 points in the inclination range. The rationale of this operation if to show that increasing the azimuthal discretization generates a solution closer and closer to the one-angle case which is averaged in the azimuth direction. Results of the computations are presented in Table 10.
### CANMOD Simulation Scene: One Angle Results

<table>
<thead>
<tr>
<th>Wavelength</th>
<th>Ang Disc</th>
<th>Tol</th>
<th>Hemispherical Ref</th>
<th>No. Iter</th>
<th>Comp Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>10</td>
<td>1.E-5</td>
<td>6.822878370432E-2</td>
<td>7</td>
<td>12.8sec</td>
</tr>
<tr>
<td>550</td>
<td>10</td>
<td>1.E-5</td>
<td>7.36905963700E-1</td>
<td>11</td>
<td>13.2Sec</td>
</tr>
<tr>
<td>650</td>
<td>10</td>
<td>1.E-5</td>
<td>7.83933825955E-2</td>
<td>8</td>
<td>12.3sec</td>
</tr>
<tr>
<td>800</td>
<td>10</td>
<td>1.E-5</td>
<td>4.359877829986E-1</td>
<td>21</td>
<td>14.2Sec</td>
</tr>
</tbody>
</table>

Table 7: Hemispherical reflectance computation via one-angle implementation.

### CANMOD Simulation Scene: Two Angle Results

<table>
<thead>
<tr>
<th>Wavelength</th>
<th>Ang Disc</th>
<th>Tol</th>
<th>Hemispherical Ref</th>
<th>No. Iter</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>10x10</td>
<td>1.E-5</td>
<td>6.853144334181E-2</td>
<td>7</td>
<td>673.1 sec</td>
</tr>
<tr>
<td>550</td>
<td>10x10</td>
<td>1.E-5</td>
<td>1.345212219403E-1</td>
<td>11</td>
<td>891.6sec</td>
</tr>
<tr>
<td>650</td>
<td>10x10</td>
<td>1.E-5</td>
<td>7.821908481941E-2</td>
<td>8</td>
<td>799.9sec</td>
</tr>
<tr>
<td>800</td>
<td>10x10</td>
<td>1.E-5</td>
<td>4.400300151475E-1</td>
<td>22</td>
<td>1141.6sec</td>
</tr>
</tbody>
</table>

Table 8: Hemispherical reflectance computation via two-angle implementation.

### CANMOD Simulation Scene: Error Analysis

<table>
<thead>
<tr>
<th>Wavelength</th>
<th>Hemispherical Ref</th>
<th>Hemispherical Ref</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>450</td>
<td>6.822878370432E-2</td>
<td>6.853144334181E-2</td>
<td>3.026596374899E-4</td>
</tr>
<tr>
<td>550</td>
<td>1.336905963700E-1</td>
<td>1.345212219403E-1</td>
<td>8.306255703000E-4</td>
</tr>
<tr>
<td>650</td>
<td>7.83933825955E-2</td>
<td>7.821908481941E-2</td>
<td>3.797465598600E-4</td>
</tr>
<tr>
<td>800</td>
<td>4.359877829986E-1</td>
<td>4.400300151475E-1</td>
<td>4.042232148899E-3</td>
</tr>
</tbody>
</table>

Table 9: Comparison of one-angle and two angle results. Error analysis.

### CANMOD 1-angle versus 2-angle: 800 nm

<table>
<thead>
<tr>
<th>Two-Angle Discretization</th>
<th>Hemispherical Ref 2- Angle</th>
<th>Hemispherical Ref 1-Angle (fixed)</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>10x10</td>
<td>4.40030E-1</td>
<td>4.35987E-1</td>
<td>4.0429999999E-3</td>
</tr>
<tr>
<td>10x12</td>
<td>4.37685E-1</td>
<td>4.35987E-1</td>
<td>1.6979999999E-3</td>
</tr>
<tr>
<td>10x14</td>
<td>4.36181E-1</td>
<td>4.35987E-1</td>
<td>1.9399999999E-4</td>
</tr>
<tr>
<td>10x16</td>
<td>4.35992E-1</td>
<td>4.35987E-1</td>
<td>4.9999999999E-6</td>
</tr>
</tbody>
</table>

Table 10: Effect of the azimuth on hemispherical reflectance computation.
| Radiant Intensity TOC: \( \theta = \text{Inclination}, \phi = \text{Azimuth} \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| \( \theta = 13.1^\circ \) | \( \theta = 30.1^\circ \) | \( \theta = 47.2^\circ \) | \( \theta = 64.3^\circ \) | \( \theta = 81.4^\circ \) |
| \( \phi = 357.7^\circ \) | 1.6593e-2 | 2.3832e-2 | 2.5449e-2 | 2.5808e-2 | 2.5856e-2 |
| \( \phi = 340.3^\circ \) | 1.6459e-2 | 2.3915e-2 | 2.5519e-2 | 2.5851e-2 | 2.5873e-2 |
| \( \phi = 307.2^\circ \) | 1.6791e-2 | 2.4063e-2 | 2.5621e-2 | 2.5914e-2 | 2.5900e-2 |
| \( \phi = 261.7^\circ \) | 1.6847e-2 | 2.4099e-2 | 2.5650e-2 | 2.5932e-2 | 2.5907e-2 |
| \( \phi = 208.1^\circ \) | 1.6718e-2 | 2.3898e-2 | 2.5508e-2 | 2.5845e-2 | 2.5871e-2 |
| \( \phi = 151.8^\circ \) | 1.6370e-2 | 2.3583e-2 | 2.5281e-2 | 2.5707e-2 | 2.5813e-2 |
| \( \phi = 98.2^\circ \) | 1.6139e-2 | 2.3435e-2 | 2.5172e-2 | 2.5641e-2 | 2.5786e-2 |
| \( \phi = 52.7^\circ \) | 1.6249e-2 | 2.3540e-2 | 2.5238e-2 | 2.5680e-2 | 2.5802e-2 |
| \( \phi = 19.6^\circ \) | 1.6167e-2 | 2.3689e-2 | 2.5358e-2 | 2.5752e-2 | 2.5832e-2 |
| \( \phi = 2.2^\circ \) | 1.6552e-2 | 2.3808e-2 | 2.5430e-2 | 2.5797e-2 | 2.5851e-2 |

Table 11: Radiant intensity at the top-of-the-canopy as function of azimuth and inclination (Planophile case).

The two-angle canopy algorithm computed hemispherical reflectances that seem to converge toward the value obtained with the one-angle formulation. Indeed, the absolute error goes down as the number of azimuthal points is increased.

The algorithm is able to compute the radiant intensity within and top of the canopy. In the 10x10 angular discretization, the radiant intensity TOC is computed for 50 points (upper hemisphere), i.e., 5 in inclination and 10 in azimuth (the rest of the points are associated to the angular rays in the lower hemisphere and are directed toward the soil). Table 11 shows the radiant intensity distribution for 10 azimuthal angles and 5 inclination angles.

The same information can be displayed using polar plots. Figure 30 shows the polar plot of the radiant intensity as function of azimuth with inclination as parameter.
Figure 30: Radiant intensity polar plot for the sun at the zenith and planophile canopy architecture. The various lines correspond to different inclination angles.

In this particular canopy scenario the effect of the azimuth is not strong. This is due to the particular architecture chosen (Planophile) and to the sun angle (Zenith). Since the leaves have higher probability to be distributed horizontally, the sunlight coming from the zenith will have the tendency to be distributed isotropically in the azimuthal direction. The effect can be seen after the third significant digit. To highlight the azimuth effect, a different canopy architecture was considered. As a matter of fact, we chose the
extremophile type whose leaves are distributed mainly horizontally and vertically. The sun inclination was set to be 13 degree while the sun azimuth was set to be 274 degree. The CANMOD two-angle code was run considering 10 inclination angles and 20 azimuthal angles. The results are reported in Table 12 where the intensity TOC is shown for the five inclination angles with \( \mu > 0 \) (photons coming out of the canopy) and 20 azimuthal angles. Figure 31 shows the polar plot of the data reported in table 12 with inclination angle as parameter. Figure 32 and Figure 33 report the polar plot for four distinct inclinations. The radiant intensity now has a strong dependence on the azimuth. We note that the radiance reached its maximum in the direction of the sun (274 degree in azimuth) which is the evidence of the hot spot. The highest anisotropic behavior is shown for the inclination of 13 degree.
<table>
<thead>
<tr>
<th>$\theta$ = Inclination, $\varphi$ = Azimuth</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 13.1^\circ$</td>
</tr>
<tr>
<td>$\varphi = 359.4^\circ$</td>
</tr>
<tr>
<td>$\varphi = 355.0^\circ$</td>
</tr>
<tr>
<td>$\varphi = 346.2^\circ$</td>
</tr>
<tr>
<td>$\varphi = 333.4^\circ$</td>
</tr>
<tr>
<td>$\varphi = 316.8^\circ$</td>
</tr>
<tr>
<td>$\varphi = 296.9^\circ$</td>
</tr>
<tr>
<td>$\varphi = 274.0^\circ$</td>
</tr>
<tr>
<td>$\varphi = 248.8^\circ$</td>
</tr>
<tr>
<td>$\varphi = 222.0^\circ$</td>
</tr>
<tr>
<td>$\varphi = 194.1^\circ$</td>
</tr>
<tr>
<td>$\varphi = 165.8^\circ$</td>
</tr>
<tr>
<td>$\varphi = 137.9^\circ$</td>
</tr>
<tr>
<td>$\varphi = 111.1^\circ$</td>
</tr>
<tr>
<td>$\varphi = 85.9^\circ$</td>
</tr>
<tr>
<td>$\varphi = 63.0^\circ$</td>
</tr>
<tr>
<td>$\varphi = 43.1^\circ$</td>
</tr>
<tr>
<td>$\varphi = 26.5^\circ$</td>
</tr>
<tr>
<td>$\varphi = 13.5^\circ$</td>
</tr>
<tr>
<td>$\varphi = 4.9^\circ$</td>
</tr>
<tr>
<td>$\varphi = 0.5^\circ$</td>
</tr>
</tbody>
</table>

Table 12: Radiant intensity at the top-of-the-canopy as function of azimuth and inclination (Extremophile case).
Figure 31: Radiant intensity polar plot for the extremophile canopy architecture. The various lines correspond to different inclination angles.

Figure 32: Radiant intensity polar plot for the 13 degree inclination angle output and extremophile case.
Figure 33: Radiant intensity polar plot for all possible inclination angles at the top-of-the-canopy in the extremophile case.
5. PART IV: LEAF-CANOPY MODEL BASED NEURAL NETWORK.

SOLUTION TO THE INVERSE PROBLEM

5.1. Introduction

The Leaf-Canopy radiative transfer model was numerically integrated to compute, for any given wavelength, the hemispherical reflectance Top-Of-the-Canopy. This procedure is usually labeled as the solution of the forward problem, i.e., given the canopy biochemical and morphological parameters as well as sun angle and soil reflectance, the radiative regime within the canopy structure can be computed. Nevertheless, in the field of remote sensing of vegetation, it is of paramount importance to solve the inverse problem: given the radiant intensity or reflectance collected by satellite and/or airborne sensors to find the biophysical canopy parameters. Therefore, if we aim to devise a model-based tool for gathering information about any canopy field, the inverse problem must be approached and solved in order to retrieve canopy biophysical parameters.

The problem of retrieving canopy biophysical parameters using radiance collected at the sensor is an extremely important topic in the area of satellite optical-NIR remote sensing. For example, all models aiming at simulating the exchange of carbon between the atmosphere and the terrestrial surface include Leaf Area Index (LAI) and fraction of Photosynthetic Active Radiation (fPAR) as state variables [Sellers, Ref 30]. Those parameters can be estimated using satellite and/or airborne remote sensing data. The philosophy underlying the retrieval process is based on the assumption that radiance
collected by sensors at different wavelengths and in different directions can be analyzed to extract information about the down-looking canopy field. One of the most appealing and interesting ways of approaching the problem, which has received a great amount of attention, is to use models based on physical principles built to simulate and compute the radiative regime within and top of the canopy. This class of models can be sub-divided in categories including radiative transfer models, geometrical-optical models and computer simulation models. In the realm of radiative transfer models, various approaches have been followed ranging from the construction of simple non-linear models, to simple two-stream models, to complex 3-D radiative transfer models including leaf optical properties and some leaf biochemical information. The aforementioned models have the common basis of trying to compute the radiative regime in a vegetation canopy and therefore explore the relationship between canopy biophysical parameters and photons radiative intensity. Because of the models' ability to represent the functional map between reflected light and canopy morphology and/or biochemistry, the models can be inverted to provide effective tools for estimating the canopy parameters. One of the major challenges in the area of optical remote sensing of vegetation is to provide operational algorithms that retrieve canopy variables both accurately and efficiently. If the operational algorithm is based on a radiative transfer model then two conditions must be satisfied: One, the radiative transfer model must capture the essential characteristic of the interaction of light and vegetation. Two, the inversion procedure must be accurate and stable (as well as efficient) enough to guarantee correct retrieval for a large variety of canopy conditions. If the radiative transfer model has undergone verification and
validation, i.e., it is a reliable model, most of the pressure is transferred to the inversion procedure which must be carefully handled.

The inversion procedure is usually the "weakest link" since it is exposed to harmful ill-conditioning and noise sensitivity. Various techniques have been developed to solve the inverse problem and have comprehensively been addressed by Kimes [Ref 21]. Standard techniques, including modified simplex method, quasi-Newton algorithms etc, are normally based on defining a cost function and finding the canopy parameters that optimize it. If N parameters have to be retrieved the problem is reduced to an optimal search in the appropriate N-dimensional space. As any minimization procedure, these methods have some difficulties in achieving globally stable optimal results.

Neural network techniques are usually considered non-standard methods and have received much attention in the past decade. Various authors attempted neural inversion of radiative transfer models. Abdelgadir [Ref 1] used a multi-layered neural network to invert the Li and Strahler model. This geometric-optical model considers the geometry and spatial distribution of trees and shrubs as input parameters to generate bidirectional reflectance. Inverse neural network modeling was used to retrieve three canopy parameters (Density, Crown ratio and height ratio) with accuracy defined by the R value of 0.85, 0.75 and 0.75 respectively. Gong [Ref 17] used the same technique to invert the Li–Strahler algorithm for the coupled atmospheric canopy model (CAC). Bidirectional reflectance was fed to the network for independent and coupled retrieval of canopy biophysical parameters (LAI, Chlorophyll, soil reflectance and canopy architecture). The relative error was found to be in the range 1-5% for independent retrieval and 17% for
simultaneous retrieval. Kimes et al. (2002) also approached the problem of retrieving canopy parameters (Forest cover, LAI and soil parameter) via neural network inversion of the 3-D DART model. While the main concern was comparing the neural network retrieval with the modified simplex technique, the performance analysis showed a Root Mean Square Error (RMSE) value of 0.021 for forest cover, 0.21 for LAI and 0.11 for soil parameter. These results are extremely encouraging because they show that neural networks can be designed and trained to learn the map between reflectance and canopy biophysical parameters. Comparison with standard techniques such as modified simplex method also shows better performances of the neural inversion methodology especially when data are not corrupted by noise.

Although previous studies indicate that neural network inversion has the potential to be more accurate than conventional inversion methodologies, we think that accuracy in retrieving canopy parameters using neural network has not been pushed to its limit. Part of the reason is related to the fact that networks are usually trained in a supervised manner which consists in feeding the designed neural architecture with input-output pairs representing the function (i.e., the model) that we want the network to learn. The more points available to represent the map, the more accurately the network learns. Indeed, while inverting physically based models (in particular radiative transfer models), the idea is to generate the training points using the model itself in a forward mode: The points will be collected and fed to a network with model output being now network input and model input now being network output (inverse map). A close look at the aforementioned studies reveals a common denominator: the training set is relatively small. This might be
due to the fact that robust radiative transfer modeling requires the inclusion of a large number of physical parameter and numerical simulation could be extremely expensive. For example Kimes [Ref 20] generated 324 forest simulations (training points) and took approximately 12 hrs of computational time. The training set was set to 4096 points with the additional points generated via function fitting. Although the results were satisfactory, we suspect that the retrieval accuracy is limited by the training set dimension. Gong [Ref 17] used only 10 training points for LAI independent retrieval, and 1620 points for the retrieval of five simultaneous canopy parameters.

In this chapter, we present the neural network inversion of the Leaf-Canopy radiative transport Model. The LCM2 forward version considered is the 1-D, one-angle, time independent radiative transfer model that embeds the essential physics dominating the interaction between light and vegetation. As previously shown, it is based on the numerical solution of the linearized Boltzmann equation for photons traveling within and between leaves. It is numerically efficient and extremely fast (it can generate hemispherical reflectance for a single canopy scene at three wavelength in 0.25 sec). The model is used to generate the training set for a network capable of retrieving canopy biophysical parameters such us LAI and chlorophyll. The goal is to show that LCM2 can be inverted to retrieve canopy parameters and it will serve as basis for the devising a tool for ripeness estimation using images coming from UAVs (section 6). The algorithm stability and efficiency are exploited to push the accuracy of the neural networks to the limit by generating increasing larger and larger training set. One of the goals is also to show the impact of having fast algorithms in improving the accuracy of retrieval.
5.2. Multilayer Neural network using back-propagation: Theory

NNs are broadly used in remote sensing from classification to inverse problem solution [Smith J., Ref 32, Abdelgadir Ref 1]. NNs, biologically inspired, are made of elements, called “neurons” connected in a variety of ways depending upon the problem to be solved. Given a set, containing input/output pairs representative of a continuous function, NNs can be trained to “learn” the underlying relationship.

Figure 34 shows a typical multilayer neural network. It is made of three layers and its structure can be symbolically represented by $R - S^1 - S^2 - S^3$, where $R$ is number of inputs and $S^1, S^2, S^3$ are the number of neurons in each hidden layer. In general, a neural network takes the input, multiplies it by a weight matrix $W$ and adds the bias vector $b$. The sum is processed by an activation function and the result passed as input to the second layer where the operation is repeated with different weights and biases. For example, the output of the first layer is represented symbolically by:

$$a^1 = f^1(W^1 p + b^1)$$  \hspace{1cm} (5.1)

While the output of the network can be represented by:

$$a = a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b_1) + b_2) + b_3).$$  \hspace{1cm} (5.2)
A multilayer neural network can be trained to approximate any function in the sense of uniform convergence, provided that the function exists and is continuous. The weights and the biases are the free parameters that can be changed during the learning procedure. If the training set is defined (i.e., input/output pairs representative of the functional relationship are available), the learning procedure consists in minimizing a definite performance index which is a function of weights and biases. Once the index is defined, the optimal weight and biases are such that the performance index is the minimum value. The chosen performance index is the Mean Square Error (MSE) defined as:

\[
F(\chi) = F(W, b) = E[e^2] = E[(t - a)^2] = E[e^T e]
\]  

(5.3)

Where \(\chi = [W, b]\) are the weights and biases to be found, \(t\) is the target (desired) output, \(a\) is the output of the network and \(e\) is the error. Since the input/output pairs are fed to the network in a sequential mode, \(E\) represents the average of the square error. Usually, this quantity is hard to compute. Therefore, the performance index is approximated by:

\[
\hat{F}(\chi) = e^T e(k) = (t(k) - a(k))^T (t(k) - a(k))
\]  

(5.4)

According to the Widroff-Hoff procedure, the product of the error vector at each iteration \(k\) is a good approximation of the average error. Starting from an initial guess for weights and biases, an optimization procedure is implemented to find the minimum of the performance index. At each iteration \(k\), the full
set of input/output pair is fed to the network ("batch mode") to find a new point in the weight space which represents the progress toward the optimal point. Different algorithms are available for searching for the minimum of a function. The simplest algorithm is the steepest descent method, but faster training rules are in general desirable. For our purpose, a suitable choice was the Levenberg-Marquardt routine. It is a derivative of the Quasi-Newton method in which the basic step can be written as follows:

$$x_{k+1} = x_k - A_k^{-1}g_k$$  \hspace{1cm} (5.5)

Here, $A_k$ is the Hessian matrix of the performance index (quadratic form) at the current value ($k$) of the weights and biases. In the Levenberg-Marquardt algorithm the Hessian matrix can be approximated by computing the Jacobian matrix containing the derivatives of the network errors with respect to the weights and biases. Usually, the Jacobian can be easily calculated by the standard back-propagation technique. The advantage is obviously related to the fact that there is no need to compute the second derivative of the performance index. The final updating rule is:

$$A = J^T J$$
$$g = J^T e$$
$$x_{k+1} = x_k - (J^T J - mI)^{-1} J e_k.$$  \hspace{1cm} (5.6)
The parameter $m$ is introduced to take advantage of both Quasi-Newton and steepest descent methods. For $m = 0$, we have the Newton method while for $m$ large we have the steepest descent. Usually the former is faster and more accurate near the minimum. The goal is to reduce $m$ at each iterate while the minimum is approached.

Figure 34: Example of multilayer neural network

5.3. Independent retrieval of canopy variables: Methods and results

The leaf-canopy nested model is first used in the forward mode to generate training sets representing input-output pairs of the desired functional relationship reflectance-canopy variables. The first series of designed networks will be able to retrieve
independently one canopy parameter. The biophysical parameters chosen for retrieval are LAI and chlorophyll content. The last variable is one of the parameter that defines the biochemistry of the single leaf for the canopy under consideration and it is a fundamental quantity used in modeling the radiative regime within-leaf. It is very well known that a-priori information dramatically improves the network retrieval capability and indeed, the best accuracy is obtained when just one parameter retrieval is attempted. A-priori information is provided by previous knowledge of any canopy parameters involved into the model: this information can be assumed known via an external source (e.g., ground experiments). The methodology used in designing, training and testing the neural network for independent retrieval can be therefore outlined:

- LCM2 is run in the forward mode to simulate a variety of canopies. If one-parameter retrieval is attempted, the set of simulated canopies will change only that parameter while the rest of variables involved in the model algorithm will be assumed known and will stay fixed. For each canopy, hemispherical reflectance is computed at four wavelengths (450, 550, 670, 750 nm) and the input-output pairs are stored as part of the training set.

- Once the training set is generated, various network architectures are produced to provide a suitable prime network candidate. Inspection of the forward map can drive the process of choosing number of layers and/or neurons. Nevertheless, the network will be able to accept four hemispherical reflectances at the abovementioned wavelengths and output, once trained, the desired variable.
The training is performed according to the methodology outlined in the previous section. Data are pre-processed to ensure efficient training. Post-processing analysis will show the network training performances.

The network will now be tested on sample-canopies. R-value and Root Mean Square Error (RMSE) will be tabulated to evaluate the accuracy of the retrieval.

It is our goal to show that network design and training can be thought of in a way that improves the accuracy of the retrieval of the single-canopy parameter. We show that training set dimension plays a key role in neural network accurate retrieval. The training set is obviously connected to the ability of the implemented canopy model to quickly and efficiently simulate canopy scenes.

5.3.1. LAI independent retrieval: design, training and testing results

According to the scheme outlined in the previous section, neural networks for LAI retrieval are designed, trained and tested. The study focuses on showing how accuracy can be improved by generating larger and larger training sets. Table 13 shows the parameters chosen for canopy simulation maintained fixed during the training set generations (a-priori knowledge). LCM2 is run in the forward mode by varying LAI in the range [0, 9]. Six training sets are generated by gradually increasing the number of training points. The LAI range is divided into equally spaced points and the number of points augmented. The first set has 20 simulation points and the other five have 100, 500, 1000, 1500 and 2000 respectively. The design process starts with deciding the particular
neural network architecture. It is very well known that there is no standard procedure to
define the number of hidden layers and the number of neurons per layer. Generally
speaking, the overall architecture depends on the shape of the function to be learned.
More specifically, the level of non-linearity plays a key role because the more non-linear
the map is, the more neurons and/or layers are required (the network needs more
computational power to represent a highly complex function). It is also known that the
ratio number of neurons/training points is important in influencing the generalization
capability of the network (ability to learn outside the training set). In most application the
nature of the map is completely unknown. In our case, the transfer model can give us an
indication of the forward map. Figure 35 shows the reflectance at four bands (450, 550,
670,750 nm) as function of LAI as generated by LCM2. The saturation phenomenon is
the only source of non-linearity and experience tells us that a 3 layer network with less
than 10 total neurons is sufficient. For our design, we choose a 4-5-1 network with four
inputs, 5 neurons with sigmoid-log activation functions, and one output linear layer.
The training session starts with data pre-processing. To make the training more efficient
the input and the target of the network are scaled such that the mean and the standard
deviation are normalized. The training set is therefore divided into two sets: 50% of the
data are used for proper training while the rest of the data are employed for validation.
The validation set used in the training session has not the same values of LAI used in
testing the network. The idea is to monitor the network behavior while the training is in
progress. The goal is to verify that the network is learning the map at points where it is
not trained (generalization). The validation set is divided into two independent sets which
are monitored independently to guarantee the function has been represented in a uniform manner. The networks are trained in batch mode, i.e., the full training set is presented to the network at each epoch (iteration) to optimize the Jacobian and Hessian evaluation typical of the Marquardt-Levenberg routine. The training procedure aims to find the global minimum of the MSE function. The "early stopping" technique is also applied: The behavior of the MSE at validation points is monitored and the training is stopped if, at any epoch, any increase of the MSE in validation points is observed. Moreover, we set 1500 as maximum number of epochs and we look for MSE values less than $10^{-3}$.

Six different networks were generated to retrieve LAI using training sets with increasing number of training points (20, 100, 500, 1000, 1500, and 2000 training points). Figure 36 shows the MSE as function of time during the training procedure. Table 13 summarizes the results of the training. Training was relatively fast (less than 111 sec worst case scenario) and the MSE function was brought to a value $O(10^{-6})$. We notice that in the case of 20 points training set the error function goes to $10^{-5}$ at points where the network is trained, but perform badly at validation points. In the other cases, MSE behaves in the same manner both in training and validation points.

After the training session is concluded, post-processing analysis is carried out to analyze the network training performances. Regression analyses between the networks response and the target is therefore conducted. The linear fit is compared against the best possible fit (straight line at 45 deg) and the R-value is shown (Figure 37). The regression analysis is performed both on training and validation targets.
Figure 35: Reflectance as function of LAI with wavelength as parameter. The plot was generated using LCM2
Figure 36: Training session results. MSE as function of number of epochs for the six networks trained on larger and larger training sets
Figure 37: Neural network post-processing analysis. Linear fit.
<table>
<thead>
<tr>
<th>Training Points</th>
<th>20</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN Architecture</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
</tr>
<tr>
<td>Epochs</td>
<td>78</td>
<td>1295</td>
<td>1500</td>
<td>1349</td>
<td>453</td>
<td>1500</td>
</tr>
<tr>
<td>Training Time (sec)</td>
<td>9.43</td>
<td>86.25</td>
<td>101.7</td>
<td>101.4</td>
<td>25.24</td>
<td>111.15</td>
</tr>
<tr>
<td>Performance Error</td>
<td>1.79e-6</td>
<td>5.29e-7</td>
<td>1.19e-6</td>
<td>1.24e-6</td>
<td>1.37e-6</td>
<td>1.28e-6</td>
</tr>
<tr>
<td>Early Stopping</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 13: Training results summary.

Regression analysis shows an R-value of 1 for each case, and a linear fit that approximate the best linear fit with a deviation oscillating from $10^{-2}$ to $10^{-5}$. These values indicate satisfactory network training for each of the considered cases.

The networks are tested against a set of chosen simulated canopy to study the accuracy of the retrieval. Table 14 shows the LAI of the chosen simulated canopies and the network retrieval results. Table 15 and Table 16 show the absolute and relative retrieval error. A closer look at the error values shows that the retrieval was successful. Accuracy generally depends on the number of points used to train the network. For example the 20-point network is able to retrieve LAI with accuracy less then 1% if LAI is greater than one. The reason is related to the fact that the map was poorly represented in the range 0-1.

Increasing the number of points always increase the accuracy of the network. In our case, for networks with training set greater or equal to 500, the retrieval accuracy is less than 0.25% in the worst case scenario (LAI = 0.255). It is also noticed that the 500-point network performs slightly better than networks with higher training points. This fact indicates that there might be a number of training points above which the network is not able to dramatically improve the retrieval capabilities. Table 17 shows the RMSE value of the LAI retrieval.
Simulated Canopy LAI Neural Network LAI Retrieval

<table>
<thead>
<tr>
<th>Simulated Canopy LAI</th>
<th>20</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2550</td>
<td>0.1353</td>
<td>0.2491</td>
<td>0.2553</td>
<td>0.2495</td>
<td>0.2500</td>
<td>0.2559</td>
</tr>
<tr>
<td>0.4160</td>
<td>0.3032</td>
<td>0.4197</td>
<td>0.4162</td>
<td>0.4192</td>
<td>0.4190</td>
<td>0.4153</td>
</tr>
<tr>
<td>0.5250</td>
<td>0.4340</td>
<td>0.5266</td>
<td>0.5250</td>
<td>0.5266</td>
<td>0.5261</td>
<td>0.5244</td>
</tr>
<tr>
<td>1.3890</td>
<td>1.4140</td>
<td>1.3900</td>
<td>1.3893</td>
<td>1.3895</td>
<td>1.3894</td>
<td>1.3887</td>
</tr>
<tr>
<td>2.5770</td>
<td>2.5747</td>
<td>2.5756</td>
<td>2.5767</td>
<td>2.5762</td>
<td>2.5771</td>
<td>2.5767</td>
</tr>
<tr>
<td>5.5730</td>
<td>5.5725</td>
<td>5.5738</td>
<td>5.5742</td>
<td>5.5744</td>
<td>5.5737</td>
<td>5.5741</td>
</tr>
<tr>
<td>7.5180</td>
<td>7.5348</td>
<td>7.5160</td>
<td>7.5179</td>
<td>7.5172</td>
<td>7.5174</td>
<td>7.5173</td>
</tr>
<tr>
<td>7.8340</td>
<td>7.8473</td>
<td>7.8343</td>
<td>7.8375</td>
<td>7.8371</td>
<td>7.8380</td>
<td>7.8376</td>
</tr>
<tr>
<td>7.9580</td>
<td>7.9604</td>
<td>7.9546</td>
<td>7.9575</td>
<td>7.9571</td>
<td>7.9583</td>
<td>7.9581</td>
</tr>
<tr>
<td>8.1900</td>
<td>8.1658</td>
<td>8.1807</td>
<td>8.1826</td>
<td>8.1823</td>
<td>8.1840</td>
<td>8.1837</td>
</tr>
<tr>
<td>8.5270</td>
<td>8.4640</td>
<td>8.5337</td>
<td>8.5311</td>
<td>8.5308</td>
<td>8.5327</td>
<td>8.5319</td>
</tr>
</tbody>
</table>

Table 14: Canopy scene simulation and LAI retrieval results

| Neural Network Retrieval: Absolute Error |
|------------------|----|-----|-----|------|------|------|
| 20    | 100 | 500 | 1000 | 1500 | 2000 |
| 0.1197 | 0.0059 | 0.0003 | 0.0055 | 0.0050 | 0.0009 |
| 0.1128 | 0.0037 | 0.0002 | 0.0032 | 0.0030 | 0.0007 |
| 0.0910 | 0.0016 | 0.0000 | 0.0016 | 0.0011 | 0.0006 |
| 0.0250 | 0.0010 | 0.0003 | 0.0005 | 0.0004 | 0.0003 |
| 0.0023 | 0.0014 | 0.0003 | 0.0008 | 0.0001 | 0.0003 |
| 0.0033 | 0.0021 | 0.0021 | 0.0018 | 0.0021 | 0.0023 |
| 0.0050 | 0.0018 | 0.0016 | 0.0015 | 0.0012 | 0.0003 |
| 0.0012 | 0.0009 | 0.0001 | 0.0003 | 0.0003 | 0.0007 |
| 0.0070 | 0.0025 | 0.0029 | 0.0029 | 0.0024 | 0.0029 |
| 0.0005 | 0.0008 | 0.0012 | 0.0014 | 0.0007 | 0.0011 |
| 0.0008 | 0.0049 | 0.0039 | 0.0034 | 0.0044 | 0.0038 |
| 0.0168 | 0.0020 | 0.0001 | 0.0008 | 0.0006 | 0.0007 |
| 0.0133 | 0.0003 | 0.0035 | 0.0031 | 0.0040 | 0.0036 |
| 0.0024 | 0.0034 | 0.0005 | 0.0009 | 0.0003 | 0.0001 |
| 0.0242 | 0.0093 | 0.0074 | 0.0077 | 0.0060 | 0.0063 |
| 0.0630 | 0.0067 | 0.0041 | 0.0038 | 0.0057 | 0.0049 |

Table 15: LAI network retrieval error for the LCM2 simulated canopies
| Neural Network LAI Retrieval: Relative Error Percentage |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 20                             | 100             | 500             | 1000            | 1500            | 2000            |
| 46.9316                        | 2.3088          | 0.1363          | 2.1390          | 1.9722          | 0.3417          |
| 27.1171                        | 0.9012          | 0.0365          | 0.7672          | 0.7325          | 0.1601          |
| 17.3306                        | 0.2967          | 0.0074          | 0.3073          | 0.2184          | 0.1216          |
| 1.7971                         | 0.0698          | 0.0224          | 0.0353          | 0.0254          | 0.0251          |
| 0.0874                         | 0.0562          | 0.0136          | 0.0294          | 0.0056          | 0.0134          |
| 0.0919                         | 0.0598          | 0.0593          | 0.0511          | 0.0599          | 0.0641          |
| 0.1309                         | 0.0465          | 0.0415          | 0.0385          | 0.0320          | 0.0072          |
| 0.0260                         | 0.0195          | 0.0023          | 0.0073          | 0.0070          | 0.0147          |
| 0.1430                         | 0.0505          | 0.0603          | 0.0586          | 0.0482          | 0.0586          |
| 0.0099                         | 0.0148          | 0.0209          | 0.0256          | 0.0119          | 0.0199          |
| 0.0127                         | 0.0748          | 0.0594          | 0.0520          | 0.0677          | 0.0583          |
| 0.2235                         | 0.0261          | 0.0020          | 0.0109          | 0.0073          | 0.0089          |
| 0.1701                         | 0.0039          | 0.0452          | 0.0400          | 0.0515          | 0.0459          |
| 0.0305                         | 0.0432          | 0.0059          | 0.0109          | 0.0040          | 0.0008          |
| 0.2957                         | 0.1139          | 0.0899          | 0.0938          | 0.0733          | 0.0774          |
| 0.7387                         | 0.0787          | 0.0476          | 0.0451          | 0.0668          | 0.0575          |

Table 16: Relative error percentage for LAI retrieval

| Root Mean Square Error (RMSE)          |
|----------------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 20                                     | 100             | 500             | 1000            | 1500            | 2000            |
| 0.0507                                 | 0.0039          | 0.0027          | 0.0031          | 0.0031          | 0.0026          |

Table 17: Root Mean Square Error comparison.

5.3.2. Independent chlorophyll retrieval: Design, training and testing results

The procedure used in retrieving the chlorophyll parameter follows the steps outlined in the previous section and will not be repeated. Six different training sets were generated via running LCM2 in forward mode. The idea is to perform a study on the effect of the size of the training set on the retrieval accuracy. The number of points in the training set is increased according to the scheme outlined in the previous section. Inspection of the map in the forward mode shows a weak level of non-linearity. The six networks are chosen to have the same architecture 4-5-1 with reflectance input at four
different wavelengths (450, 550, 670, 750 nm), 5 neurons in the hidden layer (sigmoid-log activation function) and one linear neuron in the output layer. The training procedure is repeated for the set generated to retrieve chlorophyll. Results are shown in Figure 38 and Table 18. Post processing analysis is performed via linear regression analysis. The R-values and the linear fits show that the training is satisfactory and that each network successfully learned its own training set.

<table>
<thead>
<tr>
<th>Training Points</th>
<th>20</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN Architecture</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
<td>4-5-1</td>
</tr>
<tr>
<td>Epochs</td>
<td>10</td>
<td>407</td>
<td>1500</td>
<td>626</td>
<td>1500</td>
<td>626</td>
</tr>
<tr>
<td>Training Time (sec)</td>
<td>1.5</td>
<td>26.5</td>
<td>98.6</td>
<td>78.7</td>
<td>101.2</td>
<td>88.15</td>
</tr>
<tr>
<td>Performance Error</td>
<td>1.e-22</td>
<td>6.56e-8</td>
<td>7.6e-8</td>
<td>7.53e-8</td>
<td>7.43e-8</td>
<td>7.64e-8</td>
</tr>
<tr>
<td>Early Stopping</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 18: Summary of training results for the chlorophyll case
Figure 38: Training session results for the chlorophyll retrieval design. MSE as function of number of epochs for the six networks trained on larger and larger training sets.
Figure 39: Post-processing analysis for the chlorophyll case. Linear fit.
Tests are performed to study the accuracy of the network retrieval as function of the training set size. Table 19 shows 16 canopy scenes. Each scene has a different amount of chlorophyll. The retrieval is performed with 6 networks trained with increased number of points. Table 19 shows the network retrievals for comparison with the baseline canopies. Table 20 and Table 21 report the absolute and relative error, respectively. For the 20-point neural network, performances are better than the correspondent LAI case. Indeed, the chlorophyll function is almost linear and the error is uniformly distributed throughout the range. The relative error percentage is always less than 2%.

<table>
<thead>
<tr>
<th>Simulated Canopy Chl</th>
<th>Neural Network Chlorophyll Retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td>22.5200</td>
<td>22.2514</td>
</tr>
<tr>
<td>27.6400</td>
<td>27.5525</td>
</tr>
<tr>
<td>33.3300</td>
<td>33.4517</td>
</tr>
<tr>
<td>38.9100</td>
<td>38.7095</td>
</tr>
<tr>
<td>42.1100</td>
<td>42.0463</td>
</tr>
<tr>
<td>45.1600</td>
<td>45.1294</td>
</tr>
<tr>
<td>51.9000</td>
<td>51.9271</td>
</tr>
<tr>
<td>54.2800</td>
<td>54.0821</td>
</tr>
<tr>
<td>63.1400</td>
<td>63.1509</td>
</tr>
<tr>
<td>69.0700</td>
<td>69.0548</td>
</tr>
<tr>
<td>72.4400</td>
<td>72.4866</td>
</tr>
<tr>
<td>77.6100</td>
<td>77.6694</td>
</tr>
<tr>
<td>83.7200</td>
<td>83.7394</td>
</tr>
<tr>
<td>86.1900</td>
<td>86.2127</td>
</tr>
<tr>
<td>91.5500</td>
<td>91.4318</td>
</tr>
<tr>
<td>95.3800</td>
<td>95.2135</td>
</tr>
</tbody>
</table>

Table 19: Canopy scene simulation and chlorophyll retrieval results
For the other cases, the relative error percentage oscillates between $10^{-2}$ and $10^{-4}$ % which is extremely low, compared with typical errors found in the literature [Ref 17. Ref 20]. Table 22 reports the RMSE for every case under consideration.

<table>
<thead>
<tr>
<th>Neural Network Retrieval: Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>0.2686</td>
</tr>
<tr>
<td>0.0875</td>
</tr>
<tr>
<td>0.1217</td>
</tr>
<tr>
<td>0.2005</td>
</tr>
<tr>
<td>0.0637</td>
</tr>
<tr>
<td>0.0306</td>
</tr>
<tr>
<td>0.0271</td>
</tr>
<tr>
<td>0.1979</td>
</tr>
<tr>
<td>0.0109</td>
</tr>
<tr>
<td>0.0152</td>
</tr>
<tr>
<td>0.0466</td>
</tr>
<tr>
<td>0.0594</td>
</tr>
<tr>
<td>0.0194</td>
</tr>
<tr>
<td>0.0227</td>
</tr>
<tr>
<td>0.1182</td>
</tr>
<tr>
<td>0.1665</td>
</tr>
</tbody>
</table>

Table 20: Chlorophyll network retrieval error for the LCM2 simulated canopies
### Neural Network Chlorophyll Retrieval: Relative Error Percentage

<table>
<thead>
<tr>
<th></th>
<th>20</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1928</td>
<td>0.0729</td>
<td>0.0026</td>
<td>0.0064</td>
<td>0.0062</td>
<td>0.0018</td>
<td></td>
</tr>
<tr>
<td>0.3165</td>
<td>0.0127</td>
<td>0.0104</td>
<td>0.0073</td>
<td>0.0203</td>
<td>0.0164</td>
<td></td>
</tr>
<tr>
<td>0.3652</td>
<td>0.0148</td>
<td>0.0000</td>
<td>0.0057</td>
<td>0.0124</td>
<td>0.0106</td>
<td></td>
</tr>
<tr>
<td>0.5154</td>
<td>0.0009</td>
<td>0.0209</td>
<td>0.0149</td>
<td>0.0127</td>
<td>0.0133</td>
<td></td>
</tr>
<tr>
<td>0.1512</td>
<td>0.0112</td>
<td>0.0054</td>
<td>0.0098</td>
<td>0.0150</td>
<td>0.0089</td>
<td></td>
</tr>
<tr>
<td>0.0677</td>
<td>0.0145</td>
<td>0.0196</td>
<td>0.0188</td>
<td>0.0200</td>
<td>0.0230</td>
<td></td>
</tr>
<tr>
<td>0.0522</td>
<td>0.0059</td>
<td>0.0056</td>
<td>0.0061</td>
<td>0.0035</td>
<td>0.0068</td>
<td></td>
</tr>
<tr>
<td>0.3646</td>
<td>0.0076</td>
<td>0.0079</td>
<td>0.0103</td>
<td>0.0084</td>
<td>0.0105</td>
<td></td>
</tr>
<tr>
<td>0.0173</td>
<td>0.0045</td>
<td>0.0103</td>
<td>0.0162</td>
<td>0.0101</td>
<td>0.0093</td>
<td></td>
</tr>
<tr>
<td>0.0220</td>
<td>0.0047</td>
<td>0.0104</td>
<td>0.0121</td>
<td>0.0111</td>
<td>0.0101</td>
<td></td>
</tr>
<tr>
<td>0.0643</td>
<td>0.0030</td>
<td>0.0025</td>
<td>0.0020</td>
<td>0.0018</td>
<td>0.0003</td>
<td></td>
</tr>
<tr>
<td>0.0765</td>
<td>0.0144</td>
<td>0.0142</td>
<td>0.0173</td>
<td>0.0193</td>
<td>0.0168</td>
<td></td>
</tr>
<tr>
<td>0.0232</td>
<td>0.0038</td>
<td>0.0052</td>
<td>0.0039</td>
<td>0.0047</td>
<td>0.0050</td>
<td></td>
</tr>
<tr>
<td>0.0264</td>
<td>0.0011</td>
<td>0.0016</td>
<td>0.0023</td>
<td>0.0024</td>
<td>0.0033</td>
<td></td>
</tr>
<tr>
<td>0.1291</td>
<td>0.0276</td>
<td>0.0212</td>
<td>0.0196</td>
<td>0.0169</td>
<td>0.0189</td>
<td></td>
</tr>
<tr>
<td>0.1745</td>
<td>0.0106</td>
<td>0.0029</td>
<td>3.73e-5</td>
<td>0.0012</td>
<td>0.0004</td>
<td></td>
</tr>
</tbody>
</table>

Table 21: Relative error percentage for chlorophyll retrieval

### Root Mean Square Error (RMSE)

<table>
<thead>
<tr>
<th></th>
<th>20</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1195</td>
<td>0.0090</td>
<td>0.0071</td>
<td>0.0071</td>
<td>0.0071</td>
<td>0.0071</td>
<td></td>
</tr>
</tbody>
</table>

Table 22: RMSE comparison for chlorophyll retrieval.
5.4. Neural Network inversion: Sensitivity analysis

The previous sections showed that independent retrieval of canopy biophysical parameters is possible using the model-based neural network technique. The accuracy was shown to be dependent on the training set’s dimension. Accuracy also depends on the a-priori information available. Although multiple parameter retrieval is in general possible, it is very well known that neural networks do not perform very well for simultaneous canopy element estimation [Combal, Ref 6].

Solving the inverse problem is always problematic and a lot of care must be exercised. Indeed, the inverse problem is naturally ill-posed: the solution might not be unique and small changes in the input might cause large variations in the output. The problem of inverting a physically based radiative transfer model can be, generally speaking, defined by the inversion of the following equation:

\[ \text{Ref} = \text{MOD} (\text{Conf}, P) \]  

(5.7)

Where MOD formally represents the relationship between reflectance (\text{Ref}) and parameters to be retrieved (\text{P}). \text{Conf} indicates the presence of other parameters assumed to be a-priori information. In practical applications, this equation must be inclusive of an additional term:

\[ \text{Ref} = \text{MOD} (\text{Conf}, P) + \text{Err} \]  

(5.8)
Here, \( \text{Err} \) is an additional element included to indicate all sources of errors such as modeling and measurement errors. Estimation of these errors is usually extremely hard and might play a big role in making the inversion problem ill-conditioned. The neural network inversion of LCM2 was successful but some analysis of the network stability must be performed to address the modeling and measurement error effect on the procedure.

The goal of this section is to assess the sensitivity of the neural network inversion procedure while model and measurement uncertainties are considered. Some analysis on the stability of the network algorithms can be performed by training neural networks on corrupted data and analyzing the retrieval capabilities in estimating LAI and chlorophyll both independently and simultaneously. The idea is to collect the training set via LCM2 using appropriate a-priori information and then to corrupt the set with some definite random noise. We therefore use the corrupted set to train a set of neural networks designed to retrieve LAI and chlorophyll and explore their retrieval capabilities. We want to show that the network is fairly stable and efficient when these types of errors are included into the pictures.

### 5.4.1 Sensitivity analysis: Independent retrieval of LAI and chlorophyll

The major question regarding the stability analysis of the neural network inversion procedure deals with modeling errors. Every model is based on simplifying assumptions and therefore only partial aspects of the complex physical phenomenon one wants to describe mathematically are included. Thus, errors are accepted as an inevitable part of
the modeling process. Our aim is to understand how sensitive is the LCM2 neural
network inversion with respect to model errors and uncertainties. The basic idea in
analyzing the error effects on the inversion algorithm is to corrupt the data set used to
train the neural network, train a new network on those corrupted data and test the
capability of the algorithm in retrieving one or more canopy parameters. Retrieval errors
will provide us with a measurement of the robustness of the procedure.

Two categories of neural networks, LAI and chlorophyll independent retrieval, were
considered first. Both network categories have four reflectance inputs (450, 550, 670 and
850 nm) and one output (LAI or chlorophyll). The training sets were generated by
running LCM2 in the forward mode according to the scheme outlined in the previous
sections. This time, the data were corrupted before the training session. The corruption
was performed introducing random and biased noise. Two level of corruption for the
random noise were considered (2.5% and 5% of the original reflectance value) and a 2
% biased noise. The level of noise was increased to explore the effect of additional noise
on the training procedure. The training points chosen to be representative of the
functional map, were 100 and 160 for LAI and chlorophyll respectively. For both
network categories, the architecture was chosen to be 4-30-1. The number of neurons in
the hidden layer was significantly increased to deal with the fact that usually noise
introduces a higher level of non-linearity on the functional map. Figure 40 shows the
training session for the LAI network. The goal for the MSE function was originally set to
be $10^{-4}$ but after various trials, it became clear that we were not able to push the error to
more than $10^{-5}$. This fact is related to the high non-linearity introduced by the random
noise: to reproduce the map exactly we would need a more powerful network with a greater number of neurons in the hidden layer. Nevertheless, augmenting the number of neurons increases the degree of freedom of the network (i.e., the number of weights and biases to be chosen) degrading the ability of the algorithm to generalize well. Figure 41 shows the post-regression analysis with $R = 1$ and best fit within $10^{-3}$. Similar results are obtained for the chlorophyll case.

Figure 40: Training session on corrupted data for LAI retrieval.
To test the stability of the networks trained under corrupted data, we chose 18 canopy “experiments”: LCM2 was run to generate 18 different canopy scenes varying LAI, chlorophyll and soil reflectance as indicated in Table 23. The experiments were suggested by Combal [Ref 6] for the stability analysis of a different radiative transfer model. We chose the same canopy scene mainly to have a comparison of the stability level of our neural algorithm with those published results. Soil reflectance was set at two levels (0.3 and 0.5) to simulate dry and wet soil. Other parameters were fixed. Sixteen networks were designed and trained on corrupted data to simulate the canopy fields. The outputs obtained using those networks were analyzed and compared with the true results.
<table>
<thead>
<tr>
<th>Number of Experiments</th>
<th>LAI</th>
<th>Chlorophyll</th>
<th>Soil Reflectance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>30</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1.64</td>
<td>30</td>
<td>0.5</td>
</tr>
<tr>
<td>3</td>
<td>3.01</td>
<td>30</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>6.25</td>
<td>30</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>6</td>
<td>0.86</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>1.64</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>8</td>
<td>2.34</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>9</td>
<td>3.01</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>6.25</td>
<td>50</td>
<td>0.5</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>70</td>
<td>0.5</td>
</tr>
<tr>
<td>12</td>
<td>1.64</td>
<td>70</td>
<td>0.5</td>
</tr>
<tr>
<td>13</td>
<td>3.01</td>
<td>70</td>
<td>0.5</td>
</tr>
<tr>
<td>14</td>
<td>6.25</td>
<td>70</td>
<td>0.5</td>
</tr>
<tr>
<td>15</td>
<td>0.25</td>
<td>50</td>
<td>0.3</td>
</tr>
<tr>
<td>16</td>
<td>1.64</td>
<td>50</td>
<td>0.3</td>
</tr>
<tr>
<td>17</td>
<td>3.01</td>
<td>50</td>
<td>0.3</td>
</tr>
<tr>
<td>18</td>
<td>6.25</td>
<td>50</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 23: 18 canopy scenes for neural network, trained on corrupted data, performance analysis.

Figure 42 and Figure 43 show a comparison between the true value (red circle) and the neural network prediction (blue line) for retrieving the LAI in all 18 experiments. Figure 42 is representative of the retrieval attempted using 4 of the 16 networks corrupted with 2.5% noise while Figure 43 uses 4 networks corrupted with 5% noise. The predicted LAI behavior follows fairly closely the true LAI value. This is an indication of the stability of the networks when model errors are introduced.

The same analysis was performed for the chlorophyll retrieval case. 12 out of 16 trained networks were required to retrieve chlorophyll for all possible 18 cases. Figure 44 is representative of the chlorophyll retrieval using networks corrupted with 2.5% random
noise and 2 % bias. Again, red circle is the actual value and the blue is the retrieved one.

Figure 45 represent the results for the 5% random noise addition in the chlorophyll case.

Figure 42: Comparison between LAI true value (red circles) and LAI retrieved values (blue line). Corruption noise level was 2.5%.
Figure 43: Comparison between LAI true value (red circles) and LAI retrieved values (blue line). Corruption noise level was 5%.

Figure 44: Comparison between chlorophyll true value (red circles) and chlorophyll retrieved values (blue line). Corruption noise level was 2.5%.
One way to obtain a figure of merit of the network performance is to evaluate the RMSE for LAI and chlorophyll independent retrieval at the two noise levels. Table 24 illustrates the results and compares them with similar experiments available in the literature. Combal [Ref 6] used corrupted data (2.5% relative random and 2% relative bias) to generate neural networks for SAIL 1-D radiative transfer model inversion. RMSE was evaluated for the same 18 experiments. Table 24 shows a comparison of the results.
<table>
<thead>
<tr>
<th>Root Mean Square Error Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
</tr>
<tr>
<td>SAIL 1-D 2.5% + 2%</td>
</tr>
<tr>
<td>LCM 2.5% + 2%</td>
</tr>
<tr>
<td>LCM 5% + 2%</td>
</tr>
<tr>
<td>LAI</td>
</tr>
<tr>
<td>Chlorophyll</td>
</tr>
</tbody>
</table>

Table 24: RMSE comparison with data available from the literature.

As we see from the RMSE behavior table, the RMSE is better than the one found in the literature. This is an indication of the network stability and shows that the single parameter inversion is not ill-posed. Note that doubling the noise level practically doubles RMSE. Indeed, increasing the noise level in the training data deteriorates the results but small changes in noise level do not cause large change in the result.

As shown in the previous sections of this chapter, the neural network inversion technique is an efficient and stable way of inverting the LCM2 radiative transfer model and can be applied for canopy biophysical parameter retrieval. Next section will show how to design and train neural network algorithms, based on a modified version of LCM2 adapted to describe coffee plantation fields, for coffee ripeness prediction. The final neural network algorithms represent the end product of the overall process.
6. PART V: UAV "COFFEE PROJECT": MODEL-BASED NEURAL NETWORKS FOR COFFEE RIPENESS ESTIMATION

6.1. Introduction

This chapter deals with the neural network inversion of a coupled leaf-canopy model (LCM2) for coffee cherry ripeness prediction. The idea is to develop a neural network algorithm capable of processing both ground images and airborne images, coming from Unmanned Aerial Vehicles (UAVs) flying over coffee field plantations, in order to extract information about the ripeness level of the crop present in the down-looking field. Over the past few years, NASA has had great interest in exploring the economic potential of deploying UAVs as long-duration platforms equipped with high resolution imaging systems for commercial agricultural applications. In October 2002, a team in the Ecosystem Science and Technology Branch at NASA/Ames Research Center prepared and successfully flew a UAV, equipped with off-the-shelf camera systems, over a coffee plantation at Kauai (Hawaii). The idea is to help growers to find the best possible harvesting strategy. The most important information that needs to be conveyed to the growers is the percentage of ripe, unripe and overripe cherries in the field. We devised a model-based neural network algorithm that can automate the process in an intelligent way. The radiative transfer model LCM2 and its model/numerical improvements, presented in section 2 and section 3, was used to generate samples of the reflectance/cherry percentage relation. The samples were fed to a designed neural
network to learn the aforementioned relationship. After post-training performance analysis, the neural network was connected to ground and airborne images for ripeness prediction. The results are encouraging and show the potentiality of the overall approach.

6.2. LCM2 extension: Coffee cherry spectra

The coupled Leaf-Canopy radiative transfer model, as specified in chapter two, has been built to simulate the interaction between photons and vegetation at two different levels i.e., leaf and canopy level. The global output of the nested model is the hemispherical reflectance Top-Of-the-Canopy (TOC). The model can be easily extended to include, besides leaves, other phytoelements (e.g., fruits or any other crop) such that the hemispherical reflectance will include the effect of the light interacting with these extra components of the canopy. Different approaches can be followed to include the additional canopy elements. Our goal, as specified in the introductory section, is to provide a reliable model-based neural network tool for coffee cherry ripeness prediction, using images coming from UAVs. It is therefore mandatory, as a first step to construct the neural network algorithm, to have a radiative transfer model capable of simulating the response of coffee canopies, inclusive of coffee cherries, to the sunlight.

For our purpose, the coffee cherries can be included in the radiative transport model as scattering and absorbing elements via mixing their optical properties with leaves reflectance and transmittance. Figure 46 shows the spectral reflectance for the three types of cherries usually present in a coffee field, i.e., ripe, over-ripe and under-ripe cherries. Those spectra were measured experimentally in-situ by Lee Johnson (NASA Ames) and
were implemented on the model. The logical steps used to include the cherry percentages as LCM2 input are the following: LEAFMOD inverse/forward mode provides the reflectance and transmittance of the single coffee leaf with a specified level of chlorophyll and other biochemical elements. These optical quantities are usually fed to the canopy module to compute the scattering area function. Now, reflectance and transmittance are modified to include the optical properties of the various types of coffee cherries. For each coffee canopy scene, the percentage of leaves, ripe, over-ripe and under-ripe cherries are specified and the total reflectance and transmittance input to CANMOD is a weighted sum of the reflectance and transmittance of the above mentioned components. The percentage of the leaves or single coffee species is the weight used in the summation.

It is therefore clear that the percentage of coffee cherries can be considered as part of the canopy architecture and will influence the spectral hemispherical reflectance according to the amount of cherries present in the canopy under consideration. We would like to stress that the coffee cherries are considered as reflecting/absorbing elements for which no transmittance is possible.

LCM2, modified to include the coffee cherries can be used in simulating coffee canopy scenes and will serve as basis for devising a model-based neural network algorithm capable of predicting the amount of coffee cherries present in the field.
Figure 46: Coffee Cherries measured spectral reflectance
6.3. LCM2 inversion via neural network for ripeness estimation: Design and Training

The coupled Leaf-Canopy radiative transfer model, modified to include coffee cherries as absorbing-scattering elements, is usually implemented in a forward mode. If the canopy architecture, leaf biochemistry, soil reflectance, sun angle as well as percentage of ripe, over-ripe and under-ripe cherries are specified, LCM2 can be run in the forward mode to compute, at any specified wavelength, the reflectance at the Top-Of-the-Canopy. This quantity is usually collected by camera sensors in both ground and airborne images. The model can be used to compare the signal collected at the sensors but it is not able to perform direct estimation of the canopy morphological and biophysical parameters. Model inversion is required if we aim to devise a tool for percentage estimation. We will invert the model using the above mentioned neural network technique. The goal is to devise a reliable, accurate and stable neural network algorithm capable of learning the relationship between reflectance collected at different wavebands and the coffee cherry percentages. The idea is to simulate the aforementioned map via running LCM2 in forward mode to collect points representing the desired functional relationship. The collected points are therefore fed to a designed neural network for training purposes.

An analysis of the forward map reveals that the functional direct relationship can be schematically represented as follows:

\[ R_f = f_{FOR}(D_{cl}, D_r, D_\lambda, Sp, LAI, LAD, SR, RC\%, GC\%, FC\%, L\%, \lambda) \] (6.1)
Here, $L\%$, $RC\%$, $GC\%$, $YC\%$ are, respectively, the percentage of leaf, red (over-ripe), green (under-ripe), yellow (ripe), cherries present in the field. $D_{ch}, D_u, D_p, D_{lc}$, are the concentration of chlorophyll, water, cellulose and lignin in the canopy, $d_l$ is the leaf thickness, $SA$ is the sun angle, $LAI$ is the leaf area index, $LAD$ is the Leaf Angle Distribution, $SR$ is the soil reflectance, $R_\lambda$ is the reflectance and $\lambda$ is the wavelength.

The right-hand side of the equation includes all possible variables input to LCM2. With those inputs, the algorithm can compute the reflectance at a specified wavelength (left-hand side). It is evident that the number of input parameters is very large and a full scale neural inversion might be a very hard task. A good understanding for the harshness of the full scale inversion problem can be illustrated by the following considerations.

Assume we have reflectance at three bands available from the camera sensors. If the former were the only information available, the neural network must be designed to retrieve all possible parameters. The inverse map can be represented as:

\[ [L\%, RC\%, GC\%, YC\%, D_{ch}, D_u, D_p, D_{lc}, SA, LAI, LAD] = f_{inv}(R_1, R_2, R_3) \] (6.2)

A network designed for such retrieval would have 3 inputs and 12 outputs. To retrieve 12 parameters, LCM2 must be run in the forward mode to compute the reflectance at the three bands for a large amount of canopy scenes. If any of the 12 variable’s range is represented by 10 points, the number of canopy scene simulated to have a comprehensive representation of the map will be $10^{12}$. It is clear that such number is too high to attempt any full scale simulation. Indeed, it is very well known that neural networks perform
poorly when they are trained to retrieve a large number of variables [Ref 21]. The reason is associated with the fact that, in such a case, the map to be learned by the network is poorly represented and therefore the retrieval is not efficient. In general, a-priori information is required to reduce the number of variables. Variables such as sun angle, soil reflectance, canopy angle distribution etc. can be obtained by independent observation or ground measurements. This information reduces the number of parameters to be retrieved. We focuses on what we call “reduced inverse problem” in which most of the biophysical and morphological canopy parameters are fixed in a way such that the relationship learned by the network can be schematically represented as follows:

$$[L\%, RC\%, GC\%, YC\%] = f_{BP} (R_1, R_2, R_3)$$  (6.3)

or

$$[RC\%, GC\%, YC\%] = f_{BP} (R_4, R_5, R_6)$$  (6.4)

It is our goal to design and train a neural network capable of learning such functional relationship. The neural network can be schematically represented as in Figure 47. The input of the neural network will be the reflectance collected at three different bands and the output will be the percentage of leaf, ripe, over-ripe and under-ripe cherries. It is important to point out that the number of bands required to have an efficient retrieval is somewhat arbitrary and depends on the number of bands available at the camera sensors.
In our case, for both ground and aerial images, only three bands are available. Nevertheless we will show that they are sufficient for accurate neural network retrieval.

We need to be more specific when we make a distinction between airborne (aerial) and ground images. The former are images of the coffee field taken by the Kodak-Hasselbank Helios on-board camera. The ground images are images taken from the top of a side-loaded truck bin full of coffee cherries. The latter type of images will require the network to retrieve the percentage of cherries and no leaf. It is mandatory that networks designed to deal with ground images have three outputs instead of four.

![Figure 47: Neural network “black box” scheme.](image)

Once the number of inputs/outputs has been defined, the next step is to establish the network architecture and to identify the number of training points required for accurate learning. This procedure is usually a trial and error process, but some guidance can be provided. The network architecture is usually defined by specifying the number of hidden layers and the number of neurons per layer. These specifications are related to the
number of training points and to the nature of the desired functional relationship. In
general, a higher non-linear level in the function corresponds to a larger number of
neurons, (a more powerful network) but the number of neurons cannot be as the same
order of the training set because the network will not be able to generalize well. An
inspection of some part of the function in the direct mode shows that the relationship
between the reflectance and the cherry percentage is slightly non linear. This information
can be translated in a relatively low number of neurons estimated to be on the order of
ten. For accurate training, the number of points, representative of the function, was
chosen to be at least 100 times higher than the number of neurons.

We need to distinguish between two classes of networks. The first class includes neural
networks trained to retrieve cherry percentages from ground images. They have three
inputs and three outputs. The second class of networks will be designed and trained to
process airborne images where the percentage of leaf is important. These networks have
three inputs and four outputs. The bands used for the ground case are 450, 550, 670 nm
because the images under consideration are true color RGB. For the airborne case, we
considered 550, 680, and 790 nm according to the bands provided by the UAV camera
system.

For both classes of neural networks, the procedure is identical and follows the steps
outlined in section 5. First, the training set is constructed by defining input-output pairs
representing the functional relationship as above. The points representative of the
function (6.3) or (6.4) are collected by running LCM2 in forward mode. Parameters, such
as sun angle, soil reflectance, LAD etc, were assumed to be a-priory information and kept
fixed. The classes of networks were generated by simply changing the a-priori information so that each single neural network will be suitable for the application matching the specific value of the a-priori known parameters. The latter does not affect the first class of network since the networks are trained on data coming from a special class of canopies, in which there are no leaves but just a thick slab of coffee cherries. The generation of the second class of neural networks for aerial images will play a vital role in retrieval as will be explained later in the chapter.

Various coffee canopy scenes were generated by varying leaf and coffee cherries percentages and computing the hemispherical reflectance at the three considered wavelengths. The number of points required for accurate training depends on the number of variables under consideration. The ground image case requires less points than the airborne case since more points are necessary to accurately represent a function of four variables. For the first class of networks, we considered 4126 input/output pairs, while for the second class the number chosen was 22099 pairs. The simulating time for the FORTRAN77 version of LCM2, required to generate those cases, was 1301 sec and 5524 sec, first and second class respectively. The computation was performed on a PC with dual processor Amthlon 1200 MHz each and 724 MB RAM. The percentages dominating the various canopy scenes were constrained by the condition that the sum must be equal to 100%. The architectures were chosen according to the aforementioned guidelines (See section 5) and consist of 3-5-3 and 3-4-5 for the first and second class respectively.

Next the training procedure is initiated. The training set is divided into two parts: 50% is used for proper training and the other 50% for validation and testing. The networks are
trained in batch mode with the early stopping technique. The weights and biases are randomly initiated and the MSE is minimized via Levenberg-Marquardt optimization scheme. The training goal was set to be MSE equal to $10^{-8}$ or 500 epochs maximum training time.

After the optimal weights and biases are determined, post-processing analysis is performed to verify the efficiency of the designed networks. Table 48, Table 49, Table 50 and Table 51 show the results of the training of a network of the first class and the post-processing analysis. The network was trained for 500 epochs (no early stopping observed) at the end of which MSE was recorded to be $3.17e^{-8}$. The training time was 188 sec. Post-regression analysis shows a perfect fit within $10^{-7}$ which is an indication that the network accurately learned the map. Figure 52, Figure 53 and Figure 54 show the results for the training of a network belonging to the second class and illustrate the post-processing analysis. The network was trained for 500 epochs (no early stopping observed) at the end of which MSE was equal to $3.007e^{-6}$. The training time was 868 sec, higher than the first case because of the larger dimension of the training set. Post-regression analysis shows a perfect fit within $10^{-6}$. The latter is an indication that the network accurately learned the map but with less accuracy than the first case. In the next section will further analyze the network performances by simulating a coffee canopy field and attempting a coffee cherry retrieval using a trained neural network.
Figure 48: Class-one neural network training session. MSE as function of number of epochs.

Figure 49: Post-processing analysis. Linear fit for under-ripe percentage network retrieval.
Figure 50: Post-processing analysis. Linear fit for ripe percentage network retrieval.

Figure 51: Post-processing analysis. Linear fit for over-ripe percentage network retrieval.
Figure 52: Class-two neural network training session. MSE as function of number of epochs.

Figure 53: Post-processing analysis. Linear fit for leaf and under-ripe percentage network retrieval.
Figure 54: Post-processing analysis. Linear fit for ripe and over-ripe percentage network retrieval.

6.4. Ripeness estimation via neural network: Network performance and stability

Model-based neural networks, trained for ripeness estimation, are the subject of further investigation to better understand their retrieval capabilities. In this section, a coffee canopy scene is generated by LCM2 using data not included in the training set and then processed by a trained neural network. The coffee canopy scene is defined by setting LCM2 input parameters (LAI, LAD, soil reflectance, sun angle and leaf biochemistry) as well as leaf, ripe, over-ripe, under-ripe cherry percentages and computing the reflectance at three different band (550, 680, 790 nm). The computed reflectance generates an image to be processed by a network trained using the same canopy parameters (a-priori information) and the same bands.
Figure 55 shows the false color image for the 4x4 pixel canopy scene simulated by LCM2. Table 25 shows the canopy input parameters used in the simulation (The percentages of leaf/cherries will be shown in Table 26).

<table>
<thead>
<tr>
<th>Coffee-Canopy: LCM input parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chlorophyll</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>37.8</td>
</tr>
</tbody>
</table>

Table 25: Input data for the coffee-canopy simulated scene.

Figure 55: False color image of the reflectance computed via LCM2 of the coffee-canopy simulated scene.

A 3-5-4 neural network has been trained for leaf and coffee cherry percentage retrieval. According to the scheme outlined in the previous chapter, the training set was generated by running 22099 canopy scenes to accurately represent the reflectance-cherry percentage
map. The network performance was analyzed via post-processing and linear regression (results are not reported, see section 6.3).

The network is now ready to process the coffee canopy simulated scene. For each pixel, the neural network ingests the reflectance and computes the desired percentages. The results are shown in Table 26 where the real percentages are compared against the neural network predictions. The results are arranged according to the pixel coordinates of the 4x4 scene. Table 27 shows the absolute error.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Real</td>
<td>0.3654</td>
<td>0.1400</td>
<td>0.0589</td>
<td>0.4357</td>
<td>0.4983</td>
<td>0.0595</td>
<td>0.0890</td>
<td>0.3532</td>
<td>0.4906</td>
<td>0.0712</td>
<td>0.1750</td>
<td>0.2632</td>
<td>0.4435</td>
<td>0.0142</td>
<td>0.2219</td>
<td>0.3204</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>0.3655</td>
<td>0.1398</td>
<td>0.0588</td>
<td>0.4360</td>
<td>0.4977</td>
<td>0.0592</td>
<td>0.0893</td>
<td>0.3538</td>
<td>0.4904</td>
<td>0.0707</td>
<td>0.1755</td>
<td>0.2634</td>
<td>0.4425</td>
<td>0.0140</td>
<td>0.2225</td>
<td>0.3209</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 26: Real and predicted leaf and coffee-types percentage for the 4x4 canopy scene.
The tables confirm that the trained network is capable of retrieving the coffee cherry percentages as well as leaf percentage with accuracy ranging from $10^{-3}$ (worst case scenario) to $10^{-5}$. These results also agree with the behavior shown by a typical class-two neural network during the post-processing analysis.

### 6.5. Neural networks for ripeness predictions: Ground images

As described in section 6.3, two classes of neural networks were designed and trained to deal with both ground and aerial images containing coffee cherries (leaf and no-leaf). Post-training analysis shows that both networks were successfully trained to learn the desired relationship simulated by LCM2. The next step is to test the networks on
ground and airborne images to verify the predicting behavior of real situations. At this point, we are guaranteed the network has learned the model, but in ground and real images new classes of problems arise that will affect the network prediction capabilities. For example, model and measurement errors as well as exact knowledge of a-priori canopy parameters will play a crucial role in retrieving the coffee percentages. Moreover, we need to build an interface connecting the images with the networks for efficient "communication". To understand the nature of the problem, we recall that a typical RGB image consists of a collection of pixels. Each pixel is associated with three numbers representing the signal collected by the sensor at the Red, Green and Blue band. The number is usually referred to as a Digital Number (DN) and it is related to the amount of radiant energy sensed by the camera. The radiant energy is transformed into voltage and usually stored in a range 0-255. Nevertheless, the neural networks were trained to recognize reflectance which is a true physical quantity. It is apparent that an interface image-network is required if any coffee percentage retrieval is desired. The interface consists in designing a DN-reflectance transformation which takes the DNs coming from any image and provides the reflectance input to the network.

In this section we connect class-one neural networks (no leaf) to ground images to predict the amount of ripe, over-ripe and under-ripe cherries. The images under consideration will be side-loaded truck full of coffee cherries with no leaf present. These types of images are taken after the harvest and right before going into the factory. The images will give us an opportunity to test the network retrieval capabilities on situations where the cherries can be counted. Moreover, neural network designed for ground images
cherries recognition have its own value since growers can use a camera system to estimate the percentage of various cherries going into the factory for processing.

The next section illustrate the methodology used in building the DN-reflectance transformation as well as the results obtained in applying neural networks to ground images for coffee cherry percentage estimation.

6.5.1. DN-Reflectance transformation: Problems and solutions

Building the interface image-network is generally not an easy task since the transformation DN-reflectance is not known. Indeed, various approaches have been tried before finding a suitable solution. Figure 56 shows the ground image we considered for testing the neural network retrieval capabilities. It is a 500x700 pixels RGB image (670, 550 and 450 nm) taken from the top of a truck side-loaded with coffee cherries. From the image, we can recognize yellow, green and red/brown cherries representing respectively ripe, under-ripe and over-ripe cherries. The goal is to estimate the percentage of the different types of cherries present in the image via a neural network algorithm.

The idea is to run a class-one neural network on each pixel present in the image. The interface will take the three DNs associated with each pixel and provide the reflectance as input to the network. This image is the test-bed on which our network is run to verify the predictive capability.
The first attempt to build the DN-reflectance transformation highlighted the problems that usually arise in accomplishing this task. For each band (RGB), a transformation is required. From the image, we isolated three pure cherries plus the dark point (0 DN) and we computed the average DN for each band. Then, we used the pure cherry experimental data to associate the DN to the experimental reflectance and we used a cubic interpolation to connect the points. This interface provided the network with the required reflectance at the three bands. The results were catastrophic since the network gave unphysical results with percentage both negatives and higher than 100%. What happens can be illustrated with the help of Figure 57. The blue points are the projection of the training set over the three possible planes in the RGB space (RG, RB, GB) while the red points are the projection, on the same planes, of the pixels in the truck image processed via the DN-
reflectance transformation. It is apparent that the transformation provides the neural network with reflectance outside the training set: the network is forced to extrapolate and therefore the results are not reliable.

Figure 57: Transformed reflectance (red points) falling outside the training set (blue points).

The experimental data representing the spectral reflectance coming from pure cherries are important for the Leaf/Canopy model and must be incorporated into LCM2, but alone they are of no use if we want to build the desired network-image interface. It is important that the DNs are somehow related to the modeled reflectance coming from LCM2 because this is the language understood by the trained neural network. The first step is therefore to run LCM2 in the forward mode to compute the reflectance for pure ripe, pure over-ripe and pure under-ripe cherries at each band of interest. Table 28 shows the results. The DNs, coming from cherries sampled from the truck image (Figure 56), are shown in Table 29.
<table>
<thead>
<tr>
<th>Band</th>
<th>Reflectance 100% Overripe</th>
<th>Reflectance 100% Ripe</th>
<th>Reflectance 100% Under-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>450nm</td>
<td>1.1763e-2</td>
<td>2.6155e-2</td>
<td>1.6992e-2</td>
</tr>
<tr>
<td>670nm</td>
<td>6.9184e-2</td>
<td>1.1533e-1</td>
<td>3.0062e-2</td>
</tr>
</tbody>
</table>

Table 28: LCM2 computed reflectance for pure cherries

<table>
<thead>
<tr>
<th>Band</th>
<th>DN 100% Overripe</th>
<th>DN 100% Ripe</th>
<th>DN 100% Under-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>450nm</td>
<td>90</td>
<td>240</td>
<td>180</td>
</tr>
<tr>
<td>550nm</td>
<td>60</td>
<td>180</td>
<td>170</td>
</tr>
<tr>
<td>670nm</td>
<td>40</td>
<td>110</td>
<td>88</td>
</tr>
</tbody>
</table>

Table 29: Average DN values for pure cherries extracted from the image

We are now ready to couple the modeled reflectance with the correspondent DN. The idea is to have, for each of the three bands of interest, 3 points representative of the relationship. A forth point, representing the DN associated with the zero modeled reflectance (dark point), completes the picture. Table 30, Table 31 and Table 32 show those points for each band.

<table>
<thead>
<tr>
<th>Band 1 (Red)</th>
<th>DN</th>
<th>Modeled Reflectance</th>
</tr>
</thead>
<tbody>
<tr>
<td>670 nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overripe</td>
<td>90</td>
<td>6.9184e-2</td>
</tr>
<tr>
<td>Ripe</td>
<td>240</td>
<td>1.1533e-1</td>
</tr>
<tr>
<td>Under-ripe</td>
<td>180</td>
<td>3.0062e-2</td>
</tr>
<tr>
<td>Dark Point</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 30: DN values of the pure cherries are associated to the correspondent reflectance in the red band.
Table 31: DNs of the pure cherries are associated to the correspondent reflectance in the green band.

<table>
<thead>
<tr>
<th>Band 2 (Green)</th>
<th>DN</th>
<th>Modeled Reflectance</th>
</tr>
</thead>
<tbody>
<tr>
<td>550 nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overripe</td>
<td>60</td>
<td>1.3480e-2</td>
</tr>
<tr>
<td>Ripe</td>
<td>180</td>
<td>8.4699e-2</td>
</tr>
<tr>
<td>Under-ripe</td>
<td>170</td>
<td>6.6553e-2</td>
</tr>
<tr>
<td>Dark Point</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 32: DNs of the pure cherries are associated to the correspondent reflectance in the blue band.

<table>
<thead>
<tr>
<th>Band 3 (Blue)</th>
<th>DN</th>
<th>Modeled Reflectance</th>
</tr>
</thead>
<tbody>
<tr>
<td>450 nm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overripe</td>
<td>40</td>
<td>1.1763e-2</td>
</tr>
<tr>
<td>Ripe</td>
<td>110</td>
<td>2.6155e-2</td>
</tr>
<tr>
<td>Under-ripe</td>
<td>88</td>
<td>1.6992e-2</td>
</tr>
<tr>
<td>Dark Point</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

We have a total of 12 points to be used to map Reflectance-DNs for the three bands under consideration. A cubic interpolation is subsequently performed to represent the map in all possible points. Figure 58 shows the curves obtained by the cubic interpolation.
Figure 58: DN-reflectance map. Cubic interpolation at the three band of interest.

Once the map is prepared, we move to test the coupled interface-network algorithm. The image in Figure 56 is 500x700 pixels and for each pixel, a set of three DNs is extracted and made readily available to be processed by the neural network. To have more meaningful results, we decided to average the image over a 10x10 pixel field. The idea is to divide the original 500x700 field in a 50x70 field. Therefore, each pixel in the new image has DNs in each of the three bands that are an average over the underlying 10x10 pixels. Figure 59 shows the resulting blurred image. The reason for doing this averaging is that a single pixel in the image is smaller than any cherry. Therefore, running the network on such pixel would have little meaning. While averaging, the interface Reflectance-DN is invoked to estimate the correspondent reflectance. For each pixel in
the 10x10 square, the function performing the transformation is called and the reflectance computed. This procedure is repeated for the three RGB bands. The reflectance is averaged as well. For each band, the maximum and the minimum reflectance value (coming from the LCM2) is considered in order to define an appropriate interval and force the reflectance to be within the model bounds. If, for a certain DN, the transformation gives us a value outside of the interval, this value is not considered in the average. This mechanism should force the network to give consistent results. Notice that the averaged image (Figure 59) looks darker; this is due to the fact that DNs giving reflectance out of the interval were discarded.

Once the DNs are transformed into reflectance, the neural network can be run to predict the percentage of ripe, overripe and under-ripe cherries present in the image. An analysis of the results shows an unsatisfactory estimation as will be explained. Table 33, Table 34 and Table 35 show the percentage estimate for the top- left 5x5 pixels of the averaged 50x70 pixels (Figure 60)

Figure 59: Averaged truck image after considering only transformed pixels.
Figure 60: Averaged truck image. Zoom on the top-left 5x5 area.

<table>
<thead>
<tr>
<th>Over-ripe Percentage</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3315</td>
<td>0.3670</td>
<td>0.3785</td>
<td>0.3072</td>
<td>0.1394</td>
</tr>
<tr>
<td>2</td>
<td>0.4631</td>
<td>0.2531</td>
<td>0.2385</td>
<td>0.3617</td>
<td>0.2014</td>
</tr>
<tr>
<td>3</td>
<td>0.4091</td>
<td>0.3426</td>
<td>0.4447</td>
<td>0.3663</td>
<td>0.1621</td>
</tr>
<tr>
<td>4</td>
<td>0.4128</td>
<td>0.2839</td>
<td>0.4666</td>
<td>0.3672</td>
<td>0.4473</td>
</tr>
<tr>
<td>5</td>
<td>0.3299</td>
<td>0.4110</td>
<td>0.4200</td>
<td>0.2555</td>
<td>0.2346</td>
</tr>
</tbody>
</table>

Table 33: Neural over-ripe percentage estimation of the top-left 5x5 area.

<table>
<thead>
<tr>
<th>Ripe Percentage</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0035</td>
<td>0.0587</td>
<td>-0.0373</td>
<td>0.0322</td>
<td>0.0012</td>
</tr>
<tr>
<td>2</td>
<td>-0.0530</td>
<td>0.0841</td>
<td>0.0563</td>
<td>0.0240</td>
<td>-0.0010</td>
</tr>
<tr>
<td>3</td>
<td>0.0741</td>
<td>0.0370</td>
<td>0.0346</td>
<td>0.0683</td>
<td>0.0349</td>
</tr>
<tr>
<td>4</td>
<td>0.0189</td>
<td>0.0974</td>
<td>-0.0076</td>
<td>-0.0444</td>
<td>0.0061</td>
</tr>
<tr>
<td>5</td>
<td>0.0064</td>
<td>0.0747</td>
<td>-0.0223</td>
<td>-0.0008</td>
<td>-0.0477</td>
</tr>
</tbody>
</table>

Table 34: Neural ripe percentage estimation of the top-left 5x5 area.
<table>
<thead>
<tr>
<th>Under-ripe Percentage</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6650</td>
<td>0.5743</td>
<td>0.6588</td>
<td>0.6606</td>
<td>0.8594</td>
</tr>
<tr>
<td>2</td>
<td>0.5899</td>
<td>0.6628</td>
<td>0.7052</td>
<td>0.6144</td>
<td>0.7996</td>
</tr>
<tr>
<td>3</td>
<td>0.5167</td>
<td>0.6204</td>
<td>0.5207</td>
<td>0.5655</td>
<td>0.8031</td>
</tr>
<tr>
<td>4</td>
<td>0.5683</td>
<td>0.6187</td>
<td>0.5410</td>
<td>0.6772</td>
<td>0.5466</td>
</tr>
<tr>
<td>5</td>
<td>0.6637</td>
<td>0.5143</td>
<td>0.6023</td>
<td>0.7452</td>
<td>0.8130</td>
</tr>
</tbody>
</table>

Table 35: Neural under-ripe percentage estimation of the top-left 5x5 area.

The results are unsatisfactory for two reasons: the first reason is related to the fact that the network outputs negative values. These results are not physical and therefore of little use. Nevertheless, only 7 pixels out of 25 are not acceptable, so the situation is not so catastrophic. The second and most important reason is associated with the percentage values. We notice that the estimated yellow percentage is too low, ranging from 0.1% to 9%. This is not typical of the 5x5 pixels area (where there may be less yellow cherries than in other parts of the image), but it is a trend in all considered pixels.

In order to gain more insight, we tried to see what happens if we do the same operation as above (average the image, apply the transformation Ref-DNs and run the neural network), but considering a 5x7 pixel field, i.e., repeating the procedure over a 100x100 pixel field. Figure 61 shows the averaged image under consideration.
The neural network is applied to each pixel to estimate the same type of percentages. In this case we do not see any unphysical results (that is, the estimated percentages are positives they sum up to 100%) but we have an unsatisfactory estimation of the ripe percentage. The overall results can be summarized by stating that the under-ripe percentage varies between 31%-35%, the ripe percentage varies in the range 1.7%-4.1% and the overripe percentage varies between 61%-65%. Definitely, the ripe estimation is not consistent (too low) and therefore the relationship we have developed should be discarded. A deeper investigation needs to be conducted in order to define a better interface that connects the network with the image.
6.5.2. Pure cherry study and sensitivity analysis

To better understand what kind of prediction the network is able to give us, we are going to focus on pure cherries and we are going to use the same map. It is obvious that if we extract from the image a reasonable pure cherry, we would expect the network to predict a percentage consistent with the cherry under consideration. Since the percentage of ripe cherries seems to be low, we are going to concentrate our attention on a pure yellow cherry and see how the network responds to that type of input exclusively. Figure 62 shows the sampled yellow cherry we selected from the image.

This cherry is represented by a 6x8 pixels field and the top left pixel has coordinate (27, 2) in the overall 500x700 pixels image. The network is run over this cherry with the same procedure as above and the percentage estimation is shown in Table 36.
Table 36: Neural network percentage estimation for the sampled yellow cherry.

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Overripe</th>
<th>Ripe</th>
<th>Under-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000 (100 %)</td>
<td>0.4532 (45.32 %)</td>
<td>0.1398 (13.98 %)</td>
<td>0.4070 (40.70 %)</td>
</tr>
</tbody>
</table>

The results are again unsatisfactory: for the pure yellow cherry the ripe percentage prediction is too low to be considered reasonable. Let us look at the average DNs for this cherry and compare with the average DNs used to set the yellow point in our reflectance-DNs transformation Table 37.

Table 37: Sampled yellow cherry and baseline yellow cherry used in the transformation.

<table>
<thead>
<tr>
<th>DNs</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Yellow Cherry (from image)</td>
<td>223</td>
<td>192</td>
<td>104</td>
</tr>
<tr>
<td>Pure Yellow Cherry (Used for the map)</td>
<td>240</td>
<td>180</td>
<td>110</td>
</tr>
</tbody>
</table>

In building the Reflectance-DNs map, we set the values reported in the second line to correspond to the 100% ripe cherry. The DNs for the cherry under analysis are not that far from the reference so we would expect the prediction of the network to be consistent with this observation. This indicates that the response of the network is too far away from the expected values and therefore the transformation is not reliable. We need to rethink the way the reflectance is built and we need to have a better grasp on how changes in reflectance affect the neural network response. For this reason, we performed a sensitivity analysis to understand how the network reacts to perturbations in reflectance.

As previously explained, the neural network was designed to learn the biophysical relationship between reflectance and cherry percentages on the field and it was trained on
data coming from LCM2. If the network is fed with the reflectance corresponding to the LCM2 output for 100% yellow cherry, the subsequent response will be 100% ripe cherry prediction. The question is therefore the following: How does the network respond to perturbations in the reflectance corresponding to the yellow cherry?

The idea is to increase the reference reflectance of a certain amount and monitor the network response. For each band (RGB), 1% of the correspondent reflectance was computed and stored as a incremental unit. Table 38 shows the network output when the reflectance is gradually increased of 1%, 10%, 100%.

<table>
<thead>
<tr>
<th>NN Output</th>
<th>Overripe</th>
<th>Ripe</th>
<th>Under-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1% Increment</td>
<td>-0.073</td>
<td>1.0181</td>
<td>-0.0107</td>
</tr>
<tr>
<td>10% Increment</td>
<td>-0.0720</td>
<td>1.1803</td>
<td>-0.1083</td>
</tr>
<tr>
<td>100% Increment</td>
<td>-0.5994</td>
<td>2.6458</td>
<td>-1.0464</td>
</tr>
</tbody>
</table>

Table 38: Network response under reflectance increasing perturbation.

Table 39 shows the network output when the reflectance is gradually decreased of a defined percentage.

<table>
<thead>
<tr>
<th>NN Output</th>
<th>Overripe</th>
<th>Ripe</th>
<th>Under-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1% Increment</td>
<td>0.0073</td>
<td>0.9818</td>
<td>0.0109</td>
</tr>
<tr>
<td>10% Increment</td>
<td>0.0740</td>
<td>0.8180</td>
<td>0.1080</td>
</tr>
<tr>
<td>100% Increment</td>
<td>0.8249</td>
<td>-0.8634</td>
<td>1.0385</td>
</tr>
</tbody>
</table>

Table 39: Network response under reflectance decreasing perturbation.

We notice that if we increase the reflectance in the three bands, the percentage of predicted ripeness increases, while if we decrease it, the percentage of predicted ripeness decreases. We have negative values of the network response because we are at the limit of the model and perturbations in the reflectance may cause the network to extrapolate.
This analysis shows why the method we used in building the reflectance-DN interface is inappropriate and suggests a way to proceed further. As mentioned above, we related the average DNs for the pure cherries to the correspondent modeled reflectance coming from LCM2. This is not a good idea because we considered values of the reflectance that are at the limit of the data set used to train the Neural Network and the risk for the network to extrapolate is high. Moreover, the fact that an increase in reflectance produces an increase in the ripe percentage estimation can be used to build a map that successfully relate modeled reflectance and DNs coming from the image.

6.5.3. Building the interface for Reflectance-DNs: a new strategy

As shown above, we need to change strategy and the sensitivity analysis showed the way to proceed. Instead of considering the pure cherries as the fundamental points for the map, we take the main point to be a mixture made of 33% over-ripe cherries, 34% ripe cherries and 33% under-ripe cherries. We run LCM2 to get the reflectance corresponding to this mixture for the three different bands and we connect this value to a set of DNs representing the mixture. The latter are obtained by taking the DNs corresponding to the pure cherries and performing an average for each band.

Once the “central” point is defined, we need at least another two points to complete the transformation. In this case we use the pure cherries that represent the boundary of the model. As a general fact, low DNs tend to give overripe cherries (red-brown) while high DNs tend to give ripe cherries (yellow). So we can set up a pure yellow cherry to be the point corresponding to high DNs and a pure red cherry to be the point corresponding to
low DNs. The values of the modeled reflectance and corresponding DNs are taken from the previous map. The three points representing the pair reflectance-DN for each band are shown in Table 40, Table 41 and Table 42.

<table>
<thead>
<tr>
<th>Red Band</th>
<th>Reflectance</th>
<th>DN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Cherry</td>
<td>6.9184e-2</td>
<td>90</td>
</tr>
<tr>
<td>33% Mixture</td>
<td>7.0981e-2</td>
<td>170</td>
</tr>
<tr>
<td>Yellow Cherry</td>
<td>1.1533e-1</td>
<td>240</td>
</tr>
</tbody>
</table>

Table 40: Reflectance-DN at the red band.

<table>
<thead>
<tr>
<th>Green Band</th>
<th>Reflectance</th>
<th>DN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Cherry</td>
<td>1.3480e-2</td>
<td>60</td>
</tr>
<tr>
<td>33% Mixture</td>
<td>5.4167e-2</td>
<td>136</td>
</tr>
<tr>
<td>Yellow Cherry</td>
<td>8.4699e-2</td>
<td>180</td>
</tr>
</tbody>
</table>

Table 41: Reflectance-DN at the green band.

<table>
<thead>
<tr>
<th>Blue Band</th>
<th>Reflectance</th>
<th>DN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Cherry</td>
<td>1.1763e-2</td>
<td>40</td>
</tr>
<tr>
<td>33% Mixture</td>
<td>1.8229e-2</td>
<td>79</td>
</tr>
<tr>
<td>Yellow Cherry</td>
<td>2.6155e-2</td>
<td>110</td>
</tr>
</tbody>
</table>

Table 42: Reflectance-DN at the blue band.

For each band, the three points are connected by a straight line (it is a spine interpolation with a first order polynomial). Figure 63 shows the transformation for red, green, and bleu band.

Summarizing, the main idea is to define a transformation such that the end points represent the limit of the model (red cherry and yellow cherry) and the 33% mixture point, the central point of the model, would also be the central point of the
transformation. Notice that we are not defining any map outside the ends point because those represent the boundaries of our model.

The next step will be to verify the network response over pure cherries extracted from the truck bin image.

![Graphs showing DN-Reflectance transformation for the three visible bands.](image)

Figure 63: DN-Reflectance transformation for the three visible bands.

6.5.4. Neural Network Test: Pure cherry predictive capability

Once the new transformation is defined, a test is mandatory to verify the network-interface behavior. A suitable test is to verify how the network performs for some reasonable pure cherries extracted from the image under consideration. Figure 63 shows
the selected yellow cherry. Figure 64, Figure 65 show, respectively, the selected green and red cherries. We would like to point out that each of those three cases represents only approximately a pure cherry because the nature of the image introduces extraneous information. For example in Figure 64 we see a marked change in color (dark green) near the border of the picture. This is due to the presence of shadow as well as contingent cherry of different color.

Each image representing the sampled pure cherry has been processed. For each pixel in the field, the DNs relative to RGB bands are extracted and stored. Furthermore, they are averaged over the total number of pixels to give an average DN value of the image for each band. The resulting DNs are therefore processed by the designed interface to transform the digital numbers into reflectance suitable to be the input of the network. Those values are fed to the Neural Network for cherry percentage estimation. The results are given in the Table 43.

<table>
<thead>
<tr>
<th></th>
<th>Over-ripe Prediction</th>
<th>Ripe Prediction</th>
<th>Under-ripe Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yellow Cherry</td>
<td>-0.0174 (-1.7%)</td>
<td>0.88 (88%)</td>
<td>0.1367 (13.7%)</td>
</tr>
<tr>
<td>Green Cherry</td>
<td>0.1992 (19.9%)</td>
<td>0.2106 (21.1%)</td>
<td>0.5902 (59%)</td>
</tr>
<tr>
<td>Red Cherry</td>
<td>0.7528 (75.3%)</td>
<td>0.1264 (12.7%)</td>
<td>0.1209 (12%)</td>
</tr>
</tbody>
</table>

Table 43: Network prediction over pure sampled cherries
Figure 64: Green (under-ripe) cherry sampled from the truck image.

Figure 65: Red (Over-ripe) cherry sampled from the truck image.
If we look at the yellow cherry, we see that the network predicts 88% of ripe cherry present in the image (Figure 62). This seems to be a good prediction. We do not expect 100% because some pixel in the image may have some shadow or be representative of small parts of neighboring cherries we have not been able to eliminate. Notice that in Figure 62, near the bottom right corner there is some green present; this may justify why the network predicts a 13% under-ripe. We also notice a very small (1.7%) but negative percentage for over-ripe. This means that the reflectance value is near the border of the data set used to train the neural network, so there is a small degree of extrapolation. The small negative percentage can be considered zero for our purpose, and does not affect the performance evaluation of the network significantly.

For green and red cherries, we also obtain consistent results. In both cases the network predicts a large percentage for the expected case (59% for the green cherry and 75% for the red cherry). The percentage is less than the ripe case, but it seems related to the pureness of the sampled image. Looking at Figure 64, we notice very dark pixels as well as dark yellow pixels which in turn justify the 21% ripeness prediction for the green cherry. The same behavior is observed in Figure 65 where a portion of it is green and dark yellow.

Summarizing, the tests give us positive results and the performance of the coupled network-interface can be considered satisfactory.
6.5.5. Ripeness Estimation for the side-loaded image

We are now ready to use the neural network to predict the amount of ripe, overripe, and under-ripe cherries present in the truck loaded image. The truck load is represented in Figure 56 which is the image considered in our investigation. As previously noted, we are dealing with a 500x700 pixels image and, due to the fact that the single pixel is smaller than a single cherry, an averaged image should be considered if we are interested in performing neural network sub-pixel classification. We begin by dividing the original image into a 50x70 pixels field. Each of the new pixels is an average of 10x10 old pixels. The average is performed for each band (RGB) and the resulting DNs are representative of the new pixels. Figure 66 shows the averaged image obtained by the outlined procedure. It is important to point out that not every pixel in the 10x10 fields were considered in the average. Since the DNs are going to be processed by the interface to obtain the input for the network, we have excluded any digital number that is not included in the range of the transformation. For example, if, in the red band, the DN is greater than 240 or less than 90 (see Table 40), then the pixel is not included in the average. This method ensures that the network does not extrapolate and give us consistent results. It also masks out shadows (dark pixels) and specular reflection effects (white pixels).
Every pixel in Figure 66 is processed by the interface that transforms DNs into reflectance. The latter represents the input of the network which outputs the ripe, over-ripe and under-ripe percentage estimation for each of the 3500 pixels present in the image. Table 44 shows the total percentage estimation obtained by averaging the 3500 percentages for the three cases of interest:

<table>
<thead>
<tr>
<th>Total Percentage</th>
<th>Over-ripe Percentage Estimation</th>
<th>Ripe Percentage Estimation</th>
<th>Under-ripe Percentage Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000 (100%)</td>
<td>0.3315 (33.15%)</td>
<td>0.3868 (38.68 %)</td>
<td>0.2817 (28.17 %)</td>
</tr>
</tbody>
</table>

Table 44: Neural network cherry percentages estimation.

We do not have any information about the amount of cherries contained in the side-loaded truck. Nevertheless, a validation of the neural estimation is mandatory to assess the
network performance. We decided to apply the following strategy: we considered three subsections of the original image where the amount of yellow, red and green cherries can be estimated via hand count (the number of cherries varies between 30 and 50). Each sub-image contains mostly one cherry type (yellow, green or red). The neural network is applied to those images with the same techniques as described. Thus, the neural estimation is compared against the hand count. Figure 67, Figure 68 and Figure 69 show the sub-images (mostly ripe, mostly under-ripe and mostly over-ripe respectively) while in Table 45, Table 46 and Table 47, hand counts and neural network predictions are reported for the three cases. As reported in the tables, performances are satisfactory. In the case of the mostly ripe image, the network performs worst in the under-ripe percentage prediction (4.4% error) while does best in the over-ripe case (0.9% error). In the case of the mostly under-ripe image, the network performs worst in the under-ripe percentage prediction (11.5% error) while ripe and over-ripe have almost the same error (6.8% and 4.6% respectively). In the case of mostly over-ripe image, the error ranges between 3% and 7%. It is worth noting that the network performs worst in predicting the amount of under-ripe cherries. This is probably due to the nature of the chosen transformation DN-reflectance which is, as previously explained, based on red and yellow cherries as extreme points. The green cherry is not included in the transformation and this is probably the reason why the network performance is not as good as the other cherries. Nevertheless the results are always less than 11.5% error (worst case). For most of the cases the error is less than 7%.
Figure 67: Mostly ripe sub-image.

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Count</td>
<td>23%</td>
<td>47%</td>
<td>30%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>27.4%</td>
<td>43.5%</td>
<td>29.1%</td>
</tr>
<tr>
<td>Error %</td>
<td>4.4%</td>
<td>3.5%</td>
<td>0.9%</td>
</tr>
</tbody>
</table>

Table 45: Comparison between hand count and neural network results (mostly ripe).

Figure 68: Mostly under-ripe sub-image.

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Count</td>
<td>41%</td>
<td>35%</td>
<td>24%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>29.6%</td>
<td>41.8%</td>
<td>28.6%</td>
</tr>
<tr>
<td>Error %</td>
<td>11.5%</td>
<td>6.8%</td>
<td>4.6%</td>
</tr>
</tbody>
</table>

Table 46: Comparison between hand count and neural network results (mostly under-ripe).
Figure 69: Mostly over-ripe sub-image.

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Count</td>
<td>32.4%</td>
<td>24.6%</td>
<td>40.5%</td>
</tr>
<tr>
<td>Neural Network</td>
<td>25.7%</td>
<td>31%</td>
<td>43.3%</td>
</tr>
<tr>
<td>Error%</td>
<td>6.7%</td>
<td>6.4%</td>
<td>2.8%</td>
</tr>
</tbody>
</table>

Table 47: Comparison between hand count and neural network results (mostly over-ripe).

Figure 70: Side-loaded truck image #2.
Figure 70 shows another ground image taken during the October 2002 Kauai campaign. The image is 250x250 pixels and was taken from the top of a side-loader. The ground image was processed with a class-one designed neural network for coffee cherry estimation. The goal is to perform a further validation test for the neural network and to evaluate the impact of the DN-reflectance transformation on the network retrieval. According to the interface design procedure illustrated in section 6.5.3, three pure cherry-types must be identified from the image to determine the transformation limits. This means that we require a new transformation whenever dealing with a new image for best performance. To analyze the impact of the transformation, we considered two possible interfaces. The first interface applies the new transformation to the image for best results while the second interface transforms the DNs according to the transformation defined on the previous image (Figure 56). In the first case, we say that the neural network is "tuned" on the image.

Before running the network for cherry estimation, we averaged the image according to the scheme outlined in the previous section. The results are illustrated in Figure 71 where the original DNs are averaged over 10x10 pixel scenes, for any of the three visible bands, to create a 25x25 pixels image. For each of the 625 pixels, the above mentioned two interfaces are applied to provide two sets of reflectance' as network inputs. The neural network processes the two input sets for coffee cherry percentage estimation. The results, for the "tuning" and "no-tuning" case are presented in Table 48.
Table 48: Neural network cherries estimation. Tuning and no-tuning effect.

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NN Tuning</strong></td>
<td>19%</td>
<td>25.8%</td>
<td>55.2%</td>
</tr>
<tr>
<td><strong>NN No-Tuning</strong></td>
<td>18.2%</td>
<td>21.1%</td>
<td>60.7%</td>
</tr>
<tr>
<td><strong>Difference%</strong></td>
<td>0.8%</td>
<td>4.7%</td>
<td>5.5%</td>
</tr>
</tbody>
</table>

There is not much difference between the two cases. The worst case difference is 5.5%.

To further test the results, we follow the methodology outlined in section 6.5.3. We isolate three sub-images and we perform hand counts to be compared with the network results. Figure 72, Figure 73 and Figure 74 report the chosen sub-images. Hand counts, neural network, tuning and no-tuning results are reported in Table 49, Table 50 and Table 51.
Table 49: Comparison between neural network prediction and hand count (sub-image #1).

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Count</td>
<td>11.5%</td>
<td>23.5%</td>
<td>65%</td>
</tr>
<tr>
<td>NN tuning</td>
<td>14.4%</td>
<td>21.9%</td>
<td>63.9%</td>
</tr>
<tr>
<td>NN no-tuning</td>
<td>14%</td>
<td>18%</td>
<td>68%</td>
</tr>
<tr>
<td>Error tuning</td>
<td>2.9%</td>
<td>1.6%</td>
<td>1.1%</td>
</tr>
<tr>
<td>Error no-tuning</td>
<td>2.5%</td>
<td>5.5%</td>
<td>3%</td>
</tr>
</tbody>
</table>

Table 50: Comparison between neural network prediction and hand count (sub-image #2).

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Count</td>
<td>25%</td>
<td>20%</td>
<td>45%</td>
</tr>
<tr>
<td>NN tuning</td>
<td>20.4%</td>
<td>27.3%</td>
<td>52.3%</td>
</tr>
<tr>
<td>NN no-tuning</td>
<td>20.2%</td>
<td>22.3%</td>
<td>57.5%</td>
</tr>
<tr>
<td>Error tuning</td>
<td>4.6%</td>
<td>7.3%</td>
<td>7.3%</td>
</tr>
<tr>
<td>Error no-tuning</td>
<td>4.8%</td>
<td>2.3%</td>
<td>12.5%</td>
</tr>
</tbody>
</table>
Table 51: Comparison between neural network prediction and hand count (sub-image #3).

<table>
<thead>
<tr>
<th></th>
<th>Under-ripe</th>
<th>Ripe</th>
<th>Over-ripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand Count</td>
<td>13%</td>
<td>37%</td>
<td>50%</td>
</tr>
<tr>
<td>NN tuning</td>
<td>18%</td>
<td>30%</td>
<td>52%</td>
</tr>
<tr>
<td>NN no-tuning</td>
<td>18%</td>
<td>23%</td>
<td>57%</td>
</tr>
<tr>
<td>Error tuning</td>
<td>5%</td>
<td>7%</td>
<td>2%</td>
</tr>
<tr>
<td>Error no-tuning</td>
<td>5%</td>
<td>14%</td>
<td>7%</td>
</tr>
</tbody>
</table>

The tuned neural network gives better results most of the time. The no-tuning neural network gives acceptable results most of the cases. The worst case error is 7.3% error for the tuning case and 14% for the no-tuning case. This study shows that the application of the neural network to any image does not strictly require a re-tuning of the DN-Reflectance transformation. The tuning procedure is applied for best performance.

This technique can be used to help growers to perform ground cherry counting. For example, layers of cherries can be deployed on a treadmill. A camera system could be mounted on the top of moving machine to take images of the cherries entering into the factory. The images could be processed by a neural network for cherry percentage
estimation. A no-tuning class of network will make the process independent of the image, simplifying the implementation of the overall procedure.

6.6. Neural networks for ripeness prediction: Airborne images

This section describes the methodology used in dealing with connecting the neural network algorithm to airborne images for ripeness prediction. The past section described the methodology used to test the trained networks over real images containing coffee cherries. Besides design and training, most of the efforts were directed toward building an efficient interface network-image for effective communication between the two links. It was clear that the results depended on the way the transformation DN-reflectance was designed and the overall performance is definitely influenced by the level of knowledge/understanding of the transformation. In moving toward the airborne images, a new class of problems directly related to the type of image data available arises. Figure 75 shows the airborne image we will consider in the rest of the section. It is a collage of images taken at different instants by the camera system mounted on the Helios UAV flying over the Kauai coffee field during the October 2002 NASA campaign.
Figure 75: Collage of images taken by the Helios UAV camera system. The collage partially reconstruct the Kauai coffee field.

Basically, the image is an attempt to reconstruct part of the coffee field using partial, relatively cloud-free, sector images. The image highlights the coffee field structure. Looking at Figure 75, we clearly identify ground lines which divide the ground in regions called blocks. Therefore, each well identified region marked by lines is a block. A collection of blocks is called field. Therefore, the coffee field is characterized by a collection of fields each divided in blocks. Kauai coffee company provided a map which assigned a number for both blocks and fields. From reflectance point of view, the image is a false color picture, which is a superposition of radiant energy taken at three different bands, two in the visible region and one in the NIR (580, 660 and 790 nm respectively). The original image is composed by 8-bit digital numbers. First, image processing has
been applied by scientists at NASA Ames to perform solar induced brightness correction. Second, the solar-corrected DNs were converted into reflectance using “flat-field” measurements made with a radio-spectrometer. At the end, we received the UAV image in terms of reflectance as seen in the figure with no masking to remove soil, clouds or shadow. During the time the sector images were taken, the sun zenith varied from 25 to 44 degree.

The UAV image must be processed by a class-two neural network capable of estimating the percentage of ripe, over-ripe and under-ripe as well as leaf percentage present in the field. Since reflectance is readily available we do not need to design a DN-reflectance transformation. Although this circumstance seems to simplify the job by a direct application of the neural network to the image, caution must be exercised in order to get meaningful results. One of the major problems is related to the different sources of error that affect the reflectance data. Measurement errors, modeling errors, atmospheric correction and DN-reflectance transformation errors will generally cause the reflectance to be different than the one predicted by the LCM2 model. We recall that the neural network was trained on data coming from the Leaf-Canopy model and therefore the network has learned the map reflectance-cherry percentage within those bounds. If the sum of the possible errors is large, the reflectance available to the network will lie outside the training set, forcing the network to extrapolate yielding unphysical and unreliable results. Moreover, some of the a-priori information, used to generate the appropriate training set is not available. We have some information about the sun angle range, but no information about soil reflectance or LAI or exact chemistry of the coffee leaf. These
missing pieces will definitely influence the reflectance modeled via LCM2 and therefore the training set. Here we outline the strategy followed in solving these two major problems. First, since some of the a-priori information is missing, we design various neural networks fixing the values of LAI, soil reflectance and sun angle to a value falling within a certain range. Second, we generate various training sets and train the networks accordingly. Third, we take the reflectance data coming from the airborne image and, pixel by pixel, we check if the reflectance falls inside any training set which guarantees physical results. If the reflectance falls outside all training sets, we pick the “closest” training set and we project the reflectance in such training set according to a technique we called “Domain Projection Technique” (DPT). Basically, we force the closest network to give physical results according to a specified rule.

The following sections will provide details of the implemented methodology and will show the results of the methodology as applied to the UAV image.

6.6.1. Domain Projection Technique (DPT): Outline of the method

The Domain Projection Technique (DPT) is a technique developed to deal with situations where the transformed airborne reflectance falls outside the network training set. The method consists of forcing the neural network to find a physical solution to the inverse problem by projecting the actual value of the reflectance inside the training domain. Since the neural network is a sophisticated way of interpolating data, the algorithm is able to efficiently process only data lying inside the training set, i.e.,
extrapolation is not efficient and/or reliable. If any reflectance, coming from airborne images, is located outside the domain, the neural network under consideration always outputs unphysical solutions. Consequently, the DPT is designed to project the value back into the domain so that a solution to the problem is found. The rest of the subsection is dedicated to describe this technique.

Consider a class-two neural network capable of predicting cherries percentage as well as leaf percentage. A-priori information has been considered in constructing the training set using the best possible information. From the airborne image, after appropriate image correction, reflectance is provided to the network. Before attempting any processing using the neural network let us have a visual picture of the situation. First, the domain (training set) must be identified and its boundaries defined. The domain can be described by 5 fundamental reflectance points. LCM2 is run to collect the 4 extreme cases (pure leaf, green, yellow and red cherries) which represent the corners of the domain and a fifth case (corresponding to computing the reflectance for the 25% leaf, green, yellow and red cherries) which marks the central point of the domain (central in the percentage sense but it does not necessarily correspond to the geometrical center). If the corners of the domain are set, the boundaries can be approximated by a straight line to have a visual image of the training set. The situation is illustrated in Figure 76. Here the three dimensional space is the space of the reflectance (580,660 and 790 nm). The yellow, green, red and magenta circles are the domain corners (pure yellow, green, red cherries and leaf respectively) while the black circle indicates the location of the central percentage point (25%). The approximate domain borders are defined by the straight black lines. The domain can
therefore be visualized as a tetrahedral whose shape and dimension depends on the a-
priori information used in running LCM2 for the training set definition.

![Figure 76: Training set in the reflectance space.](image)

To better explain the technique, let us assume that a set of three reflectance values are
available from an airborne image which is represented by the blue circle in Figure 76. It
is apparent that the reflectance provided by the image lies outside the training set. If the
reflectance coming from the airborne image is processed by the neural network the result
will be unphysical (e.g., negative percentages) and the domain projection technique must
be implemented. The DPT begins by evaluating the distance between the reflectance
point and the 4 pure cherries reflectance in order to determine the two minimum distance
cases. This information is stored and will be used in a second moment to correct the
projection procedure. Subsequently, the projection algorithm is applied as follows. First, the vector joining the actual reflectance point and the center of the domain is computed and normalized to define the marching direction. The reflectance is therefore moved in the computed direction according to a "marching" speed based on the distance between the reflectance point and the center and/or the current value of the network output. At each step, the new reflectance is processed by the neural network and the output evaluated. If the results are unphysical, the procedure is repeated. Eventually, the reflectance point hits the border domain giving physical percentage results. The marching speed can assume three different values (high, medium and low). We use high speed when the distance between the reflectance and the center is greater than the minimum distance between the center point and the extreme cases. At the same time the absolute value of the network response is evaluated to define the optimal speed. Medium speed is used if the distance reflectance-center point is less than the minimum cherry-center point but one of the absolute values of the network output is between 3 and 1.1 (300% and 110% which are unphysical results). The idea is that while marching toward the center the network percentage output, although unphysical, will tend toward a physical value. When one of the network output components is less than 1.1, the speed is switched to the lowest value in order to catch the domain border. Indeed, the procedure is stopped when the neural network percentage prediction is physical. There are certain situations in which the speed is not switched fast enough and the domain is missed. If the projection does not give physical percentage within a certain number of marching points, the failure mode is implemented: the reflectance is brought back to the original value and the marching
procedure is repeated with the lowest speed which now is guaranteed to hit the domain border. Figure 77 shows the sequence of the operations. The red line indicates the vector joining the actual reflectance and the central point of the domain. The blue circles are the reflectance “marching” toward the domain set. In this case, it takes 8 iterates to reach the domain. Table 52 shows the output of the neural network while the reflectance marches toward the domain (To get the percentage one must multiply by 100). Looking at the table, one can see that, for the current reflectance coming from the image (Ref 1 in Table 52), the output of the network is unphysical. The output changes while marching toward the domain set: eventually it gets physical, i.e., the reflectance hit the domain border (Ref 9 in Table 52).

Once the reflectance is projected on the border domain, the neural network gives a physical percentage prediction. The result is usually not satisfactory due to the fact that the sought percentage generally lies inside the domain. To account for this possibility, the obtained network percentage prediction must be modified. One possible scheme is the following. Consider the original reflectance coming from airborne images and evaluate the distance to the leaf and the pure cherry points. Select the closest two. The assumption is that if the reflectance is, for example, closer to a yellow cherry reflectance point, the percentage of yellow cherry should be larger. Therefore, in this specific case, we consider the vector joining the central point and the yellow cherry point and we select a point on this line and we evaluate the network response. This operation adds more yellow percentage to the overall result.
The point is now inside the domain and yields a physical result. The closer to the yellow cherry the more yellow percentage will be predicted. It is clear that the point we choose is somewhat arbitrary. For our purpose, we select points that lie 1/3, 1/2 and 2/3 of the overall distance (2/3 is the closest to the pure cherry). The second closest point is chosen
because there could be different situations. We specifically distinguished three possible cases. In the first case, called “Biased Case” the distance between the closest and second closest is more than 15%. In this case, the reflectance is projected toward the center for percentage prediction and another point on the line joining the central point and the closest cherry is selected yielding another percentage prediction. The two values are averaged and the result considered as final percentage. In the biased case we also distinguish two cases: If the distance to the second closest is larger than 30% than we select a point that stays 2/3 of the distance between pure cherry and central point (i.e., 1/3 from the pure cherry). If the distance is between 15% and 30% the point chosen lies at ⅓ of the distance. We would like to stress that the closer to one cherry with respect to the other, the higher the percentage of that cherry for this specific case.

The second case is called “Close Case” and it is the case when the difference between the closest and the second closest is less than 2%. In this situation, we project on the domain and we also consider two points that are 1/3 of the distance from the central point on the lines that connect the central point and the two closest pure cherries. The three predicted percentage are averaged for the final result.

The third case is called “Regular Case” and it covers a situation in which the distance between the reflectance and the two closest cherries is between 2% and 15%. Again the reflectance is projected on the domain for percentage estimation and the two other percentages are averaged as in the previous case (1/2 of the distance).

The adjustment performed on the pure projection is somewhat arbitrary and must be tested to show that the performances of the overall DPT are satisfactory. The next section
will illustrate the results obtained by applying the DPT to a conceptual coffee field generated by LCM2.

6.6.2. Domain Projection Technique (DPT): Testing the method

Figure 78 shows a 4x4 simulated coffee cherry field. This false color image was generated by selecting, for each pixel, the coffee cherries and leaf percentages and then running LCM2 to generate the corresponding hemispherical reflectance. The field was designed to test the DPT technique. Since every reflectance was generated by LCM2, all 16 points lie inside the training set. The idea is to randomly scatter the reflectance point outside the domain and then apply the DPT to see how close the projected points come to the original ones. To scatter the points, we considered the vector joining the original reflectance points and the central domain point and we applied a transformation consisting of a random rotation and a random stretching of the vector. The result was to obtain points (tip of the transformed vector) outside the training set. Consequently, the DPT was applied yielding the results shown in Table 53. Here, the original percentages and the neural network prediction via DPT are compared. For each single pixel, the error on a single cherry percentage prediction is never above 19%. Performances depend on the type of mixed pixel applied to the network. Eight out of sixteen pixels are biased cases with different intensity (>30% or between 15% and 30%). In those cases, we tend to do very well in all predictions with error less than 10% (See pixel (2,2), (2,3), (3,2)), with some situations where we predict 3 out of 4 cases less than 10% and one case above (See pixel (4,4)). In the regular case scenario, we usually have 2 out of 4 percentages with
prediction error less than 10% (See pixel (3, 3)). The error gets worst when the two percentage values are close although still in the regular case. The close case situations have performances similar to the regular cases. For pixel (1,2) we see that although 3 percentages have the same value, the network predicts well only two of them.

Table 53 shows the total field percentage and the network prediction as well as the total error. We see that the global performances are satisfactory. Table 54 shows the error limits of the DPT.

Figure 78: 4x4 simulated coffee canopy scene.
<table>
<thead>
<tr>
<th></th>
<th>Field Percentage</th>
<th>NN Prediction</th>
<th>Global Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Leaf %</strong></td>
<td>8.6250</td>
<td>18.4375</td>
<td>9.8125</td>
</tr>
<tr>
<td><strong>Grn %</strong></td>
<td>29.1250</td>
<td>21.6875</td>
<td>7.4375</td>
</tr>
<tr>
<td><strong>Yel %</strong></td>
<td>31.5000</td>
<td>29.2500</td>
<td>2.2500</td>
</tr>
<tr>
<td><strong>Red %</strong></td>
<td>30.7500</td>
<td>30.5625</td>
<td>0.1875</td>
</tr>
</tbody>
</table>

Table 54: Global network error.
6.6.3. Neural Network ripeness estimation: DPT application and results

The Domain Projection Technique (DPT), illustrated in the previous section, serves as a basis for building a more complex algorithm capable of analyzing the airborne image (Figure 75) and extracting the percentage of ripe, over-ripe and under-ripe cherries present in the field. The final product is therefore a "box" which has the reflectance provided by the image at three bands as input and cherries as well as leaf percentages as output. As previously mentioned (section 6.6.1) one other major problem is associated with having the correct a-priori information in order to train the appropriate class-two neural network. For example, the sun angle varies from 25 to 44 degrees during the time the collage of images was taken. It represents the only partial information available as sun angle input to LCM2. LAI as well as soil reflectance are not available a-priori. Nevertheless, those previously mentioned parameters are input to LCM2 to generate any training set and they are required to determine the best neural network for ripeness prediction. One way to approach the problem is to train not just one specific network but a set of neural networks each capable of predicting cherries and leaf percentages but all trained on sets generated with different a-priori parameters. In essence, we make a certain number of networks available and we let the algorithm pick the best possible network, according to a specified criterion.

LAI, soil reflectance and sun angle are the three basic parameters required to complete the a-priori information and run LCM2. We defined three possible values for each of those variables. For LAI, the range considered was 0-9 while the soil reflectance typically falls in the range 0-1. The sun angle is constrained by the 25-44 degree requirement.
imposed by the airborne image information. Based on these values, 27 neural networks were designed and trained. Table 55 shows the map of all possible 27 networks with their associated values of LAI, Soil Reflectance (SR) and Sun Angle (SA).

The 27 neural networks are readily available to be coupled with the DPT to provide the ripeness estimation for the airborne image (Figure 75). In principle, the scheme followed for ripeness estimation can be summarized as follows: For each pixel on the image, the actual value of the reflectance for the 580, 680, and 790 nm bands, is extracted. The 27 networks are stored together with the pure cherry points defining the domain limits as well as the central domain point (25% point) for a total of (5x27) 135 points. The distance between each of the domain points and the reflectance is computed and stored. Those distance values are compared to define the minimum distance domain. The neural network associated with the minimum distance domain is chosen as the operating network. The reflectance is processed by the chosen network and the cherries and leaf percentages computed. If no physical value is provided, the DPT is readily applied until a physical result is achieved.

This technique can be applied to the airborne coffee field image to provide blocks and field percentage. Before the direct application of the abovementioned procedure, we need to explain the type of data we have available for analyzing the performance of the overall technique.
Table 55: Table of the a-priori input parameters for generating the 27 class-two neural networks.

The Kauai Coffee Company is the largest producer of coffee in the USA. The collage of airborne images is a picture that illustrates a medium portion of the full scale coffee field owned by the company. Every year, the coffee beans are harvested in a time range that typically goes from end of September to mid-November. Because of the size of the coffee field, Kauai coffee company heavily relies on mechanical harvesting. One of the major problems faced by the company is to correctly estimate the percentage of ripe cherries in any given field. Currently, they rely on what is commonly called “branch count”: few selected coffee trees are taken out of thousands in the field and the percentage of the ripe, over-ripe and under-ripe cherries estimated. The percentages are assumed as good approximation of the real percentage in the field. It is clear that this technique can be meaningless since the ripeness of a coffee tree depends on many factors and it is not uniform both spatially and temporally. The Kauai coffee plantation was the selected site chosen by a group of scientists from NASA Ames for demonstrating the commercial
potential of UAV as alternative to the current branch count technique. Kauai coffee company also provided the NASA team with some data regarding the harvesting results. Usually, for a specific field, all varieties coffee cherries are harvested and loaded on trucks and transported to the factory for post-harvesting processing. Here, the green cherries are roughly separated from the others and weighted. Furthermore, the ripe and over-ripe cherries are washed and immersed in water. After a certain amount of time, the over-ripe cherries usually float and can therefore be separated from the ripe ones. The two cherry-types are weighted and the percentages computed. Table 56 shows the parchment data from their process provided to the NASA team by the company. Data on nine fields are provided including the percentage of ripe (R %) under-ripe (G %) and over-ripe (F %), time of the harvesting and branch count associated to the field under consideration (note that the over-ripe cherries percentage is denoted with F% meaning “floater”).

Before processing the airborne image by the neural networks and comparing to the results with the parchment data, we must identify which fields we are able to extract out of the collage. Indeed, according to the plantation map, some of the fields in Table 56 are not present in the airborne image while others are only partially recognizable due to presence of clouds and/or due to the cut in the image. Other fields are fully captured but no parchment data are available. After careful analysis of the image, it turns out that only three fields can be clearly identified: field 406, field 408 and field 410 as illustrated in Figure 79. The fields 408, 406 and 410 are composed of 4, 14 and 6 blocks respectively and can be processed by using the technique illustrated in the last section.
<table>
<thead>
<tr>
<th>FIELD</th>
<th>G%</th>
<th>R%</th>
<th>F%</th>
<th>F% +R%</th>
<th>Branch Counts</th>
<th>harvest date</th>
<th>Branch Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>320</td>
<td>63.28%</td>
<td>8.60%</td>
<td>28.12%</td>
<td>36.72%</td>
<td>harvest date</td>
<td>13-Nov</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>402</td>
<td>46.52%</td>
<td>36.03%</td>
<td>17.45%</td>
<td>53.48%</td>
<td>Branch Counts %yell</td>
<td>8-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>404</td>
<td>35.89%</td>
<td>22.89%</td>
<td>41.22%</td>
<td>64.11%</td>
<td>Branch Counts</td>
<td>5-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>406</td>
<td>52.51%</td>
<td>17.30%</td>
<td>30.19%</td>
<td>47.49%</td>
<td>Branch Counts</td>
<td>11-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>408</td>
<td>39.45%</td>
<td>39.55%</td>
<td>21.00%</td>
<td>60.55%</td>
<td>Branch Counts</td>
<td>13-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>409</td>
<td>46.03%</td>
<td>28.46%</td>
<td>25.51%</td>
<td>53.97%</td>
<td>Branch Counts</td>
<td>9-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>410</td>
<td>71.61%</td>
<td>15.14%</td>
<td>13.25%</td>
<td>28.39%</td>
<td>Branch Counts</td>
<td>3-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>412</td>
<td>27.00%</td>
<td>33.74%</td>
<td>39.26%</td>
<td>73.00%</td>
<td>Branch Counts</td>
<td>24-Oct</td>
<td>Branch Counts</td>
</tr>
<tr>
<td>416</td>
<td>67.46%</td>
<td>17.40%</td>
<td>15.14%</td>
<td>32.54%</td>
<td>Branch Counts</td>
<td>4-Oct</td>
<td>Branch Counts</td>
</tr>
</tbody>
</table>

Table 56: Parchment data (Kauai Coffee Company).
Figure 79: Position of the 406, 408 and 410 fields.

Figure 80: Field 408.
The three fields, as shown in Figure 80, Figure 81 and Figure 82 are subjected to image processing before the neural network analysis. Each field is extracted from the collage.
and the single blocks are identified and numbered according to the map provided by the growers. Before processing, a masking procedure is applied to each block to isolate every block element in the analysis. The images have one meter spatial resolution with a total of 882,365 pixels making an overall 24 blocks available. The total number of pixels to be processed is extremely high and it requires a large amount of computing resources. To speed up the procedure, we decided to consider pixel aggregates and average the reflectance over the three bands. Each block image was divided in pixel rows. Each row was divided into elements containing a sequence of 5 pixels. Every five pixels reflectance was averaged yielding, for each block, a collection of rectangular pixels 1x 5 meters. Thus, the number of pixels was reduced to 176,473 (factor of five).

The neural networks coupled with the Domain Projection Technique were then applied to the aggregated fields for ripeness estimation. Table 57, Table 58 and Table 59 show the results. Each table reports the ripeness prediction results for both blocks and associated fields. The neural networks predict the ripe, under-ripe, over-ripe and leaf percentage for every 1x5 averaged pixel. Those predictions are averaged over the number of pixels composing the block. Those results are reported in the first four columns of the tables. The fifth column shows the computational time for both blocks and fields. The last three columns focus on the pure cherry percentages. The leaf percentage is subtracted and a linear transformation is applied to obtain the percentage of the single-specie cherry relative to the total amount of cherries. The results are reported as ripe, under-ripe and over-ripe blocks and fields percentages. After each averaged pixel analysis and prediction, the estimated percentages are stored in three-dimensional matrices. That
information is used to create “ripeness maps”. For every single pixel in the block, a color scale was associated with the percentage of the predicted pure cherry. Red, yellow and green color scales were associated to the percentage of over-ripe, ripe and under-ripe, respectively. The color scheme is designed to track the following prescription: the brighter the color, the higher the percentage of the cherry-type in the block. Figure 83 and Figure 84 show the ripeness maps for two of the 24 blocks analyzed. Those figures show the cherry percentage distribution over the blocks and it is an extremely useful tool for monitoring the cherry distribution over the fields. Ripe, under-ripe and over-ripe distribution can be visually captured. A color-bar also helps the reader to associate the color with a numerical value. Growers can easily utilize those types of images to determine the best harvesting strategy and if UAV images are continuously provided over the harvesting season, ripeness time behavior can be monitored and analyzed. The rest of the ripeness maps generated via neural networks are reported in appendix A.
### Field 408

<table>
<thead>
<tr>
<th></th>
<th>Leaf %</th>
<th>Grn%</th>
<th>Yel%</th>
<th>Red%</th>
<th>Comp Time (sec)</th>
<th>Under-Ripe</th>
<th>Ripe %</th>
<th>Over-Ripe %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>52.09</td>
<td>16.79</td>
<td>11.47</td>
<td>19.63</td>
<td>1108</td>
<td>35.06</td>
<td>23.94</td>
<td>40.99</td>
</tr>
<tr>
<td>Block 2</td>
<td>54.04</td>
<td>18.39</td>
<td>10.81</td>
<td>16.75</td>
<td>1856</td>
<td>40.02</td>
<td>23.52</td>
<td>36.44</td>
</tr>
<tr>
<td>Block 3</td>
<td>61.00</td>
<td>13.17</td>
<td>9.99</td>
<td>15.82</td>
<td>1490</td>
<td>33.77</td>
<td>25.63</td>
<td>40.58</td>
</tr>
<tr>
<td>Block 4</td>
<td>61.23</td>
<td>14.11</td>
<td>10.50</td>
<td>13.80</td>
<td>11272</td>
<td>36.73</td>
<td>27.33</td>
<td>35.93</td>
</tr>
<tr>
<td>Field</td>
<td>57.09</td>
<td>15.61</td>
<td>10.69</td>
<td>16.50</td>
<td>15726</td>
<td>36.39</td>
<td>25.10</td>
<td>38.49</td>
</tr>
</tbody>
</table>

Table 57: Block neural network percentage estimation results for the coffee field 408.

### Field 406

<table>
<thead>
<tr>
<th></th>
<th>Leaf %</th>
<th>Grn%</th>
<th>Yel%</th>
<th>Red%</th>
<th>Comp Time (sec)</th>
<th>Under-Ripe</th>
<th>Ripe %</th>
<th>Over-Ripe %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>30.28</td>
<td>39.10</td>
<td>9.20</td>
<td>21.40</td>
<td>708</td>
<td>56.09</td>
<td>13.20</td>
<td>30.70</td>
</tr>
<tr>
<td>Block 2</td>
<td>29.89</td>
<td>41.18</td>
<td>9.15</td>
<td>19.76</td>
<td>330</td>
<td>58.74</td>
<td>13.05</td>
<td>28.19</td>
</tr>
<tr>
<td>Block 3</td>
<td>24.62</td>
<td>45.78</td>
<td>10.48</td>
<td>19.09</td>
<td>476</td>
<td>60.75</td>
<td>13.91</td>
<td>25.33</td>
</tr>
<tr>
<td>Block 4</td>
<td>36.22</td>
<td>31.72</td>
<td>10.69</td>
<td>21.35</td>
<td>3188</td>
<td>49.74</td>
<td>16.77</td>
<td>33.48</td>
</tr>
<tr>
<td>Block 5</td>
<td>32.95</td>
<td>33.44</td>
<td>11.16</td>
<td>22.42</td>
<td>3772</td>
<td>49.88</td>
<td>16.65</td>
<td>33.45</td>
</tr>
<tr>
<td>Block 6</td>
<td>29.80</td>
<td>33.99</td>
<td>12.88</td>
<td>18.78</td>
<td>2007</td>
<td>51.76</td>
<td>19.63</td>
<td>28.60</td>
</tr>
<tr>
<td>Block 7</td>
<td>30.47</td>
<td>31.86</td>
<td>10.57</td>
<td>19.77</td>
<td>2800</td>
<td>51.21</td>
<td>17.00</td>
<td>31.78</td>
</tr>
<tr>
<td>Block 8</td>
<td>28.02</td>
<td>35.10</td>
<td>11.24</td>
<td>20.97</td>
<td>4069</td>
<td>52.14</td>
<td>16.70</td>
<td>31.15</td>
</tr>
<tr>
<td>Block 9</td>
<td>33.63</td>
<td>28.52</td>
<td>13.40</td>
<td>17.76</td>
<td>2685</td>
<td>47.78</td>
<td>22.45</td>
<td>29.76</td>
</tr>
<tr>
<td>Block 10</td>
<td>39.69</td>
<td>19.95</td>
<td>19.54</td>
<td>11.98</td>
<td>2053</td>
<td>38.76</td>
<td>37.95</td>
<td>23.27</td>
</tr>
<tr>
<td>Block 11</td>
<td>43.30</td>
<td>14.42</td>
<td>20.16</td>
<td>13.90</td>
<td>4006</td>
<td>29.73</td>
<td>41.59</td>
<td>28.66</td>
</tr>
<tr>
<td>Block 12</td>
<td>35.24</td>
<td>25.90</td>
<td>15.04</td>
<td>17.46</td>
<td>1695</td>
<td>44.35</td>
<td>25.75</td>
<td>29.89</td>
</tr>
<tr>
<td>Block 13</td>
<td>32.30</td>
<td>31.24</td>
<td>11.15</td>
<td>19.32</td>
<td>2634</td>
<td>50.61</td>
<td>18.07</td>
<td>31.30</td>
</tr>
<tr>
<td>Block 14</td>
<td>39.44</td>
<td>25.03</td>
<td>11.33</td>
<td>17.97</td>
<td>3329</td>
<td>46.06</td>
<td>20.86</td>
<td>33.06</td>
</tr>
<tr>
<td>Field</td>
<td>33.28</td>
<td>31.23</td>
<td>12.57</td>
<td>18.71</td>
<td>33752</td>
<td>49.11</td>
<td>20.97</td>
<td>29.90</td>
</tr>
</tbody>
</table>

Table 58: Block neural network percentage estimation results for the coffee field 406.
<table>
<thead>
<tr>
<th>Block</th>
<th>Leaf%</th>
<th>Grn%</th>
<th>Yel%</th>
<th>Red%</th>
<th>Comp Time (sec)</th>
<th>Under-Ripe %</th>
<th>Ripe %</th>
<th>Over-Ripe %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>22.70</td>
<td>37.1</td>
<td>11.0</td>
<td>29.15</td>
<td>4529</td>
<td>48.04</td>
<td>14.24</td>
<td>37.71</td>
</tr>
<tr>
<td>Block 2</td>
<td>13.40</td>
<td>39.33</td>
<td>10.98</td>
<td>36.27</td>
<td>1091</td>
<td>45.42</td>
<td>12.68</td>
<td>41.89</td>
</tr>
<tr>
<td>Block 3</td>
<td>30.34</td>
<td>34.36</td>
<td>14.53</td>
<td>20.74</td>
<td>1201</td>
<td>49.33</td>
<td>20.87</td>
<td>29.78</td>
</tr>
<tr>
<td>Block 4</td>
<td>24.76</td>
<td>38.04</td>
<td>11.95</td>
<td>25.23</td>
<td>1098</td>
<td>50.56</td>
<td>15.89</td>
<td>33.53</td>
</tr>
<tr>
<td>Block 5</td>
<td>33.13</td>
<td>32.47</td>
<td>15.07</td>
<td>19.32</td>
<td>1542</td>
<td>48.55</td>
<td>22.54</td>
<td>28.90</td>
</tr>
<tr>
<td>Block 6</td>
<td>27.05</td>
<td>38.19</td>
<td>11.19</td>
<td>19.35</td>
<td>4912</td>
<td>55.56</td>
<td>16.27</td>
<td>28.15</td>
</tr>
<tr>
<td>Field</td>
<td>25.23</td>
<td>36.59</td>
<td>12.45</td>
<td>25.01</td>
<td>14373</td>
<td>49.58</td>
<td>17.08</td>
<td>33.33</td>
</tr>
</tbody>
</table>

Table 59: Block neural network percentage estimation results for the coffee field 410.

Figure 83: Field 408 Block 1
Neural network results must be compared with parchment data to estimate the prediction error. Table 60, Table 61 and Table 62 reports the parchment data available for the three fields and the neural network field results. Moreover, branch count is also reported for comparison purposes. Neural network prediction errors and branch count errors are computed and shown in the last two rows. The last column shows the sum of the ripe and over-ripe cherries as well.

The neural network prediction works extremely well in field 406. The error ranges from 0.29% to 3.67 % in any of the four columns reported in the table. Branch count performs its worst with errors ranging from 17.03% to 45.37%. Field 408 errors range from 3.06%
to 17.49% while field 410 performs the worst with errors ranging from 1.94% to 22.03%.

As shown in the tables, the networks perform, in any possible case, better than the branch count technique (field 406). For field 408 and 410, the network is able to capture one percentage with 3% and 1.94% error respectively. Also, for field 408, the sum R%+ O% stays within the 3% error. Some considerations must be made regarding the parchment data. First of all, the data are expected to be fairly noisy. Indeed, the percentages are estimated on the weight of the cherries processed by the factory. Due to the mass separation process and the floating technique used to divide ripe from over-ripe cherries, we are not expecting this data to be highly accurate. Time factor also plays a big role. Figure 75 is a collage of images taken from the UAV on September 30, 2002, while most of the harvesting was performed from mid-October to early November.

<table>
<thead>
<tr>
<th>Field 408</th>
<th>Under-ripe %</th>
<th>Ripe % (R)</th>
<th>Over-ripe % (O)</th>
<th>R + O %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parchment Data</td>
<td>39.45</td>
<td>39.55</td>
<td>21.00</td>
<td>60.55</td>
</tr>
<tr>
<td>Neural Network</td>
<td>36.39</td>
<td>25.10</td>
<td>38.49</td>
<td>63.59</td>
</tr>
<tr>
<td>Branch Count</td>
<td>x</td>
<td>21.14</td>
<td>47.08</td>
<td>68.22</td>
</tr>
<tr>
<td>Error NN %</td>
<td>3.06</td>
<td>14.45</td>
<td>17.49</td>
<td>3.04</td>
</tr>
<tr>
<td>Error Branch %</td>
<td>x</td>
<td>18.41</td>
<td>26.08</td>
<td>7.67</td>
</tr>
</tbody>
</table>

Table 60: Comparison between neural network prediction, parchment data and branch count for field 408.
In the end we would also like to report the networks effectively used in the prediction. As previously mentioned, 27 networks were designed to account for missing a-priori information. The “decision” algorithm selects one according to the minimum reflectance distance criterion. The networks used for prediction were labeled and only three networks were reported. According to Table 55, they correspond to the networks numbered 1, 10 and 19. Table 63, Table 64 and Table 65 report, for each field and block the total amount of averaged pixel processed and how many times #1, #10 and #19 networks were used. In most of the cases, each block tends to use just one type of network guaranteeing certain homogeneity in the overall neural response.
### Field 408

<table>
<thead>
<tr>
<th></th>
<th>Total Pixels</th>
<th>NN #1</th>
<th>NN #10</th>
<th>NN #19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>3046</td>
<td>1480</td>
<td>1368</td>
<td>198</td>
</tr>
<tr>
<td>Block 2</td>
<td>6734</td>
<td>3159</td>
<td>3146</td>
<td>429</td>
</tr>
<tr>
<td>Block 3</td>
<td>4888</td>
<td>1792</td>
<td>2635</td>
<td>461</td>
</tr>
<tr>
<td>Block 4</td>
<td>25055</td>
<td>8774</td>
<td>14433</td>
<td>1848</td>
</tr>
</tbody>
</table>

Table 63: Neural networks used in percentage prediction for field 408.

### Field 406

<table>
<thead>
<tr>
<th></th>
<th>Total Pixels</th>
<th>NN #1</th>
<th>NN #10</th>
<th>NN #19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>1946</td>
<td>325</td>
<td>199</td>
<td>1422</td>
</tr>
<tr>
<td>Block 2</td>
<td>907</td>
<td>74</td>
<td>81</td>
<td>752</td>
</tr>
<tr>
<td>Block 3</td>
<td>1309</td>
<td>113</td>
<td>115</td>
<td>1081</td>
</tr>
<tr>
<td>Block 4</td>
<td>8765</td>
<td>490</td>
<td>326</td>
<td>7949</td>
</tr>
<tr>
<td>Block 5</td>
<td>10369</td>
<td>1031</td>
<td>732</td>
<td>8606</td>
</tr>
<tr>
<td>Block 6</td>
<td>5517</td>
<td>284</td>
<td>384</td>
<td>4849</td>
</tr>
<tr>
<td>Block 7</td>
<td>7752</td>
<td>340</td>
<td>337</td>
<td>7075</td>
</tr>
<tr>
<td>Block 8</td>
<td>12288</td>
<td>2884</td>
<td>1296</td>
<td>8108</td>
</tr>
<tr>
<td>Block 9</td>
<td>8207</td>
<td>363</td>
<td>465</td>
<td>7379</td>
</tr>
<tr>
<td>Block 10</td>
<td>5644</td>
<td>117</td>
<td>156</td>
<td>5371</td>
</tr>
<tr>
<td>Block 11</td>
<td>11318</td>
<td>174</td>
<td>219</td>
<td>10925</td>
</tr>
<tr>
<td>Block 12</td>
<td>5211</td>
<td>532</td>
<td>415</td>
<td>4264</td>
</tr>
<tr>
<td>Block 13</td>
<td>8068</td>
<td>590</td>
<td>550</td>
<td>6928</td>
</tr>
<tr>
<td>Block 14</td>
<td>9704</td>
<td>357</td>
<td>286</td>
<td>9061</td>
</tr>
</tbody>
</table>

Table 64: Neural networks used in percentage prediction for field 406.

### Field 410

<table>
<thead>
<tr>
<th></th>
<th>Total Pixels</th>
<th>NN #1</th>
<th>NN #10</th>
<th>NN #19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>13552</td>
<td>6674</td>
<td>1005</td>
<td>5873</td>
</tr>
<tr>
<td>Block 2</td>
<td>3001</td>
<td>2600</td>
<td>139</td>
<td>262</td>
</tr>
<tr>
<td>Block 3</td>
<td>3352</td>
<td>658</td>
<td>368</td>
<td>2326</td>
</tr>
<tr>
<td>Block 4</td>
<td>2701</td>
<td>1100</td>
<td>271</td>
<td>1330</td>
</tr>
<tr>
<td>Block 5</td>
<td>3912</td>
<td>567</td>
<td>382</td>
<td>2963</td>
</tr>
<tr>
<td>Block 6</td>
<td>13227</td>
<td>3438</td>
<td>1683</td>
<td>8106</td>
</tr>
</tbody>
</table>

Table 65: Neural networks used in percentage prediction for field 410.
7. SUMMARY AND CONCLUSION

The research presented in the previous sections, which constitutes the body of the dissertation, can now be summarized. The goal of this final section is to review the fundamental points of the work and briefly highlight the methodology applied to solve the research objectives as well as the contribution given to the scientific community.

In this dissertation, we developed a methodology aimed at solving what is usually called the “end-to-end” problem in radiative transport in dense canopy plants. Starting from first principles, applied to describe the interaction between light and vegetation, we devised a coupled leaf-canopy radiative transfer model. The mathematical structure of the model was explored and the equations describing the photons transport in canopies and within leaves numerically solved. The computation, at any desired wavelength, of the radiant intensity and hemispherical reflectance completed the solution of the forward problem. Thus, the inverse problem, which is the most important in remote sensing practical applications, was approached using a neural network technique. Various neural networks were designed and trained to learn the inverse functional relationship between reflectance and structural canopy parameters. The same technique was used to train model-based neural networks capable of estimating the percentage of various types of coffee cherries and to devise an “intelligent” algorithm to help growers in defining the best harvesting strategy. The final product, or final end, of the overall procedure is
therefore a tool that can be generalized and used in the area of the crop estimation using satellite and/or airborne remote imagery.

The solution of the “end-to-end” problem required the completion of five fundamental tasks: Modeling, spectral theory, numerical implementation, neural network inversion and application to a real-life problem.

In the modeling section, we presented the methodology used to devise a coupled leaf-canopy radiative transfer model. The modeling approach was not new but aimed at extending the computational capabilities of the existing LCM2. The transport of photons was considered and analyzed at two different levels. At the leaf level, the scattering and absorption coefficients were used to define the interaction between photons and leaf interior. The balance of photons yielded LEAFMOD considered in both one-angle and two-angle formulations yielding the equations for retrieving the leaf optical properties.

At the canopy level, a one-dimensional, time independent, radiative transport model was devised to accurately describe the transport of light in canopy media (CANMOD). The typical anisotropic structure of the vegetation yielded a more complicated transport equation which was considered in both the one-angle and two-angle formulations. The previous version of LCM2 embedded the one-angle formulation obtained averaging out the effect of the azimuth. We extended the model by considering the transport of photons in all possible directions. The two-angle formulation canopy equation, together with the appropriate boundary conditions, were obtained and considered for numerical implementation.
In the spectral theory section, we considered the canopy equation in its one-angle formulation and explored its rich mathematical structure. The inscattering kernel was assumed separable of finite rank. The goal was to determine the spectrum and find the eigenfunctions and therefore to develop a "Caseology" for the canopy equation. We found that the spectrum is divided in two parts: The discrete spectrum and the continuous spectrum. The discrete spectrum consisted of M discrete eigenvalues in the complex plane, derived by setting the dispersion function equal to zero. The continuous spectrum was also found and consisted of the section $\nu \in [-\xi_m, \xi_m]$ of the real axis. The M discrete eigenfunctions and infinite singular continuous modes were determined. Orthogonality of the modes (both discrete and continuous) was established and completeness proved under the stricter assumption that the coefficients of the scattering kernel were entire functions.

In the numerical implementation section, we proceeded to numerically solve the transport equations for both leaf and canopy transport. The existing LEAFMOD and CANMOD, embedded in the original LCM2 consider only the one-angle formulation and use the FN method to determine both leaf optical properties and canopy hemispherical reflectance. First, we also considered the one-angle formulation and solved the equation using the discrete ordinate method (SN). The discretized equations were implemented on a digital computer and results compared with the FN solution. Moreover, the two angle formulation was approached with the SN method. The two-angle discrete equations were derived and implemented to compute the radiant intensity as function of the inclination and azimuthal angle. The last step greatly enhances the model computational capabilities. The numerical procedure was tested to ensure consistency and numerical accuracy.
In the neural network inversion section, the solution of the forward problem was used to generate a training set representative of the functional relationship between spectral reflectance and canopy parameters. The goal was to devise neural network algorithms for the retrieval of canopy parameters such as LAI and chlorophyll. A large amount of canopy scenes were generated to define the appropriate training set. We designed neural networks with various architectures depending on the parameters we wanted to retrieve. Design and training was successfully performed and post-processing analysis gave satisfactory network performances. The accuracy was compared with some data available in the literature. Sensitivity analysis was performed to analyze the effect of modeling uncertainties. Various networks were trained on corrupted training sets to verify the ability of the designed networks to retrieve canopy parameters. The neural networks were able to retrieve the LAI and chlorophyll parameters with reasonable errors showing good stability under perturbations.

In the last section, we considered the problem of analyzing airborne and ground images containing coffee cherries for ripeness prediction. The goal was to design two classes of neural networks capable of extracting valuable information about cherry percentages from images coming from the ground and captured by the UAVs camera system flying over coffee plantations. LCM2 was modified to include three types of coffee cherries (ripe, over-ripe and under-ripe) as scattering and absorbing (non-transmitting) elements. The model was used to generate training sets representing the functional relationship between spectral reflectance and percentage of ripe, over-ripe and under-ripe cherries. The sets were used to train various neural networks for ripeness prediction. After post-
processing and performance analysis, the two classes of networks were connected to the images. Ground images, consisting of pictures taken by ground cameras positioned on the top of side-loaded truck bins, were interfaced with the class-one networks. Since the ground images were reported as DNs, we designed an interface capable of performing the DN-reflectance transformation required to determine the RGB reflectance which is the network input. The neural network predictions were compared with cherry hand counts and the error estimated. The results were satisfactory showing 11% worst case prediction error.

Airborne images, coming from the Helios UAV imagery system, were connected to class-two neural networks. No DN-reflectance transformation was required since the data were provided as reflectance at the 580, 660 and 790 nm. The Domain Projection Technique (DPT) was developed to deal with measurement and modeling errors as well as with the lack a-priori information. The technique was designed to avoid unphysical results whenever the image reflectance falls outside of the training set (extrapolation). DPT was tested on a conceptual coffee field. Twenty-seven neural networks were designed to deal with the lack of a-priori information. After proper masking, any pixel coming from the block/field under consideration undergoes a “decision” algorithm which selects the best neural network in the sense of the minimum distance. The selection is followed by the application of the DPT technique. Three fields were processed by the networks and ripe, over-ripe and under-ripe percentage estimated. The results were compared with available parchment data and branch count. The neural networks perform better than branch count in all situations. When compared with parchment data, the error shows a bound of
approximately 20%. This technique seems to give good results although the amount of data available for validation is limited.

It is important to point out that the Domain Projection Technique was implemented to overcome the network inability to produce physical results. Indeed, the unphysical cases are not produced by the network algorithm but model and/or measurement errors. In other words the theory cannot fully account for the observation. The technique was designed to provide an empirical remedy to alleviate the problem. Although the results are acceptable while applied to airborne images, future work should focus on fixing the mismatch between the LCM2 radiative transport theory and the remote sensing data.

The final product of the overall procedure is a tool that can be readily applied to real-world problems as demonstrated in the case of the NASA coffee project. Indeed, the "intelligent" neural algorithm will be an extremely useful device in helping growers to find the best harvesting strategy. The five-step process shows how extremely complex transport theory can be translated into a very practical remote sensing instrument.

The methodology presented can be applied to different scenarios. In the field of precision agriculture, the radiative transfer model could be modified to include a large variety of crops as absorbing-scattering elements. In the same fashion, model-based neural networks could be trained to retrieve the amount of ripe crop using UAV airborne images. The potential of the method must be further explored since it could have a tremendous impact on the way agriculture is done.

The radiative transfer model, coupled with its neural network inversion technique, might have applications in fields other than agriculture. One of the possible future works is
associated with the construction of a similar radiative transfer model to simulate the interaction between light and coral structures. The model must be modified to include coral optical properties. We can imagine coral structures having the same types of canopy architecture and appropriate coral "leaf" biochemistry. The reflectance at the top of the coral structure could be computed with the techniques presented in the previous section. The neural network inverse technique could be used to monitor the health status of coral reef.

In conclusion, the methodology developed, which starts from the application of first principles and ends with providing a tool for coffee-cherry ripeness prediction, seems to have a great potential for a large variety of applications. The common denominator is the use of a radiative transfer model and its inverse neural technique to exploit the vast amount of possibilities offered by flying UAVs.
APPENDIX A

This appendix reports ripeness prediction maps, generated by the class-two neural networks, for all processed blocks.

Figure 85: Field 408 Block #3
Figure 86: Field 408 Block #4

Figure 87: Field 406 Block #1
Figure 88: Field 406 Block #2

Figure 89: Field 406 Block #3
Figure 92: Field 406 Block #6

Figure 93: Field 406 Block #7
Figure 94: Field 406 Block #8

Figure 95: Field 406 Block #9
Figure 96: Field 406 Block #10

Figure 97: Field 406 Block #11
Figure 102: Field 410 Block #2

Figure 103: Field 410 Block #3
Figure 104: Field 410 Block #4

Figure 105: Field 410 Block #5
Figure 106: Field 410 Block #6
REFERENCES


Van Kampen, N. G., Physica (Utrecht) 21, 949 (1955). [Ref 34]


