Imaging skin pathologies with polarized light: empirical and theoretical studies

Jessica C. Ramella-Roman

M.S., Electrical Engineering, University of Pavia – Italy (1993)

A dissertation submitted to the faculty of the
OGI School of Science & Engineering
at Oregon Health & Science University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
In
Electrical and Computer Engineering

January 2004

The dissertation “Imaging skin pathologies with polarized light: empirical and theoretical studies” by Jessica C. Ramella-Roman has been examined and approved by the following Examination Committee:

Steven L. Jacques
Professor
Thesis Research Advisor

Scott A. Prahl
Assistant Professor

Sean Kirkpatrick
Associate Professor

Ken Lee, MD
Assistant Professor
School of Medicine, OHSU

Monica Hinds
Assistant Professor

Dedication

To my family.

Volli, sempre volli, fortissimamente volli.

Vittorio Alfieri

Acknowledgements

I would like to express all my gratitude to the many people that contributed to the construction of this thesis. First my advisor, Dr. Steven L. Jacques that gave me the opportunity and the means to pursue a Ph.D. when I was drowning in the sea of middle management and wondering what to do with my life. Steve never stopped asking for the best I could give and for this and all his help in my scientific life I will always be grateful. I also want to thank Dr. Scott A. Prahl that fostered my growth as a scientist and as an individual. Scott extended to me his time and knowledge ever so generously. His scientific curiosity and ability to see the potential in a student make him stand out, and are for me source of admiration and example.

I also want to thank the other members of my committee: Dr. Sean Kirkpatrick for giving me a different prospective on optics and helping me find a great laboratory for my post-graduate work, Dr. Ken Lee for his time and expertise in the clinical part of this thesis, and Dr. Monica Hinds for always being available and supportive.

Special thanks goes to Dr. Michael Evans at Colorado State University, for his help and discussions on his adding doubling program; the comparison of the results of RT3 and my Monte Carlo programs was a turning point in my student career.

My gratitude goes to Dr. Paulo Bargo, that I could always turn to for a good scientific discussion, help and mutual therapy. Paulo and I went through many similar hoops, I am grateful that he went first making my life a little bit easier. I want to thank Ted Moffitt for all our discussions on science, politics and sports. Ted has been a great friend and endless supplier of scientific trivia and mac-rumors. Teresa Godell for being a caring friend and an example of student and scientist.

Many thanks to John, Kirstin, Jon, Yin-Chu, Dan and Lisa for their friendship and Rob for calling always at the right time, (Friday 2.30 pm) with the right reason.

All my gratitude goes to the friends and colleagues at the Oregon Medical Laser Center and at Oregon Graduate Institute for always handing a helping hand.

A big *grazie* goes to Dray and Brian for all their unconditional support, great dinners and many bottles of wine.

Finally I want to thank my husband Patrick for being my source of strength, and for patiently supporting me on the ups and downs of my graduate life and my parents Lella and Pietro that always accepted and encouraged their crazy daughter following her thousands of kilometers around the globe.

Contents

Dedication iii

Acknowledgements iv

Abstract xx

1 Introduction 1
1.1 Motivation 6
1.2 Skin Cancer 6
1.1.2 Squamous Cell Carcinoma 7
1.2.2 Basal Cell Carcinoma 7
1.2.3 Melanoma 8
1.3 Non-invasive diagnostic techniques 9
1.4 Invasive diagnostic techniques 10
1.5 Polarized light 10
1.6 Polarized light in medicine 11
1.7 What is polarized light 12
1.8 The Stokes vector and Mueller matrix formalism 15
1.9 Thesis goals 18

2 Design, testing and clinical studies of a handheld polarized light camera 21
2.1 Introduction 21
2.2 Materials and Methods 22
2.2.1 System Calibration 26
2.2.1.1 Camera dark noise 26
2.2.1.2 Camera repeatability 27
2.2.1.3 Polarization optics test 28
2.3 Error Analysis 31
2.4 Clinical Results 36
2.4 Discussion 42

3 Three Monte Carlo programs of polarized light transport into scattering media 43
3.1 Introduction 43
3.2 Standard Monte Carlo program 45

3.3 Method 1- Meridian Planes Monte Carlo 46
3.3.1 Launch....48
3.3.2 Move 48
3.3.3 Drop 49
3.3.4 Scatter 49
3.3.4.1 Rotation of reference frame into scattering plane....54
3.3.4.2 Scattering of the photon of an angle α in the scattering plane 54
3.3.4.3 Return of the reference frame to a new meridian plane....54
3.3.5 Photon life....56
3.3.6 Boundaries 56
3.4 Method 2 –Euler Monte Carlo 56
3.4.1 Launch....57
3.4.2 Move 57
3.4.3 Drop 57
3.4.4 Scatter 57
3.4.5 Photon life....59
3.4.6 Boundaries 59
3.5 Method 3 –Quaternion Monte Carlo....62
3.6 Rejection method 64
3.7 Results 66
3.7.1 Comparison with Evans Code....66
3.7.2 Experimental validation....70
3.7.3 Mueller Matrix of backscattered light from micro spheres solutions....73
3.8 Conclusions 77

4 Heuristic models of polarized light travel into scattering media 78
4.1 Introduction....78
4.2 Development of an heuristic model - the first χ model 79
4.3 Experimental results....84
4.4 Analysis of χ model....90
4.4.1 Modeling with Monte Carlo of the experiment, revisiting the heuristic fit..90
4.4.2 Influence f and χ in the heuristic model 92
4.4.3 The χ parameter and the de-phasing of the polarization for increasing τ.....95
4.4.4 Poincaré sphere analysis 98
4.4.5 Polarized light distribution through a slab....103
4.5 Influence of size parameter and g in polarized light transmission through a slab111
4.7 A new heuristic model of polarized light transmission through scattering media118
4.8 Application of the new heuristic model to a slab geometry....121
4.9 Conclusions....124

5 Experimental analysis of polarized light travel into biological tissues 125
5.1 Introduction....125

5.2 Reflection of a rough metal band embedded into a scattering media 126
5.2.1 Materials and methods 126
5.2.2 Results 126
5.3 Mueller Matrix description of collimated light transmission through liver, muscle and skin 137
5.3.1 Introduction 137
5.3.2 Materials and methods 138
5.3.3 Tissue preparation 140
5.3.4 Modeling of Mueller matrix polarimeter 140
5.3.5 Calibration and error analysis of the Mueller matrix polarimeter 146
5.3.6 Results 151
5.3.6 Discussion 163
5.4 Characterization of microscopic domains of birefringence in thin tissue sections163
5.4.1 Materials and methods 163
5.4.3 Conclusions 168

6 Evaluation of spherical particle sizes with an asymmetric illumination microscope170
6.1 Introduction 170
6.2 Material and methods 172
6.2.1 Image evaluation 174
6.2.2 Collection model 175
6.3 Results 178
6.4 Fitting of the data 181
6.5 Preliminary cell work 183
6.5 Discussion 185

7 General discussion and conclusions 186
7.1 Introduction 186
7.2 Construction of a polarized hand-held device for skin cancer borders determination 187
7.3 Three Monte Carlo programs of polarized light transfer into scattering media ..188
7.4 Heuristic model of polarized light transport in mono disperse solutions of micro spheres and scattering media 190
7.5 Depth of penetration of polarized light in biological tissues and macro retardation tests 191
7.6 Asymmetric microscope to study precancerous nuclei enlargement 192

Appendix A: Hand-held prototype control software 193
A.1 Apple License Agreement 193
A.1.1 Source code 193
Appendix B: Three Monte Carlo rograms 222
B.1 Meridian plane Monte Carlo 222
B.2 Euler Monte Carlo 237
B.3 Quaternions Monte Carlo 247

Appendix C: Muelle matrices of biological tissue 259
C.1 The Data Matrix recorded as transmission 259
C.1.1 Skin 259
C.1.2 Muscle 262
C.1.3 Liver 265

Bibliography 269

Biographical Note 280

List of Tables

3.1 Reflectance mode, comparison between Evans adding-doubling code and the meridian plane Monte Carlo program. The results do not include the final rotation for a single detector. 68

3.2 Transmission mode, comparison between Evans adding doubling code and the meridian plane Monte Carlo program. The results are not corrected for a single detector... 68

4.1 Fitting parameters used in the fits of figure 4.10 and 4.11... 112

5.1 Table of polarizing element positions to construct Mueller matrices. r... 142

5.2 Transformations necessary to form Mueller matrix from Data matrix... 145

5.3 Comparison in between values of birefringence available in the literature and this thesis values..... 168

List of Figures

1.1 Simple schematic of polarized light imaging. A light source (monochromatic in the figure) is oriented at an angle a to the skin normal. The light beam is polarized parallel to the source/sample/detector plane. Some of the light penetrating the skin layers is backscattered in the detector direction. This light is analyzed with an analyzer oriented sequentially parallel (PAR) and perpendicularly (PER) to the source polarization. A polarized image is reconstructed with the two PAR and PER images using equation 1.1. 2

1.2 First clinical prototype tested at OHSU Dermatology, with Dr. Ken Lee. 4

1.3 Image of a benign pigmented nevus. The left-hand side image is unpolarized, and the image on the right is the polarized image. Notice how illumination variation over the image is corrected in the right-hand side image. 4

1.4 Image of a freckle. The left-hand side image is unpolarized. The image on the right is the polarized image. The freckle disappears in the polarized image, because superficial melanin absorption is cancelled in the POL equation (1.2). 5

1.5 Image of a tattoo. The left-hand side image is unpolarized. The image on the right is the polarized image. Tattoo pigment are enhanced by the polarized camera. 5

1.6 Left-hand side is the unpolarized light image. The right-hand side is the polarized light image. Images were taken in the clinic with a manual system (fig 1.2) [3] consisting of a 16-bit digital camera and two high quality polarizers. The white areas in the right-hand side image indicate air bubbles under the glass plate in contact with the skin. 7

1.7 Basal cell carcinoma. The left image was taken with unpolarized light. The right image was taken with polarized light. The system used for these images was that of figure 1.2 [2]. The white areas in the right-hand side image indicate air bubbles under the glass plate in contact with the skin. 8

1.8 Image of a melanoma *in situ*. The right-hand side image was taken with a polarized handheld device described in chapter 2. On the left-hand side is the image captured with parallel horizontal detector and source polarizers. 9

1.9 The $E_{\parallel}$ and $E_{\perp}$ components of the total electric field **E** are shown aligned with the x and y axes, respectively. The angle ϕ indicates the orientation of the E field at a moment in time, viewed as the wave approaches the observer. The wave is propagating along the z-axis toward the lower left toward the observer. 13

1.10 The electric fields for linear H, V, P+ and P- polarized light and for circular R and L polarized light.....15

2.1 The main components of the hand-held polarization system: polarizing beam-splitter cube, two CCD cameras and an objective lens. The CCD cameras are attached to a Delrin support enclosing the beam-splitter cube.....23

2.2 The clinical hand-held polarization system, spherical casing protects the main components and makes the prototype easy to hold with one hand... 24

2.3 Dark noise experiment. The average value of a dark noise image in laboratory units is plotted versus acquisition time.....27

2.4 Performance test for polarized optics. The x-axis denotes the angle of orientation of the linear polarization of the light source (0° denotes parallel to horizontal axis of beamsplitter). The y-axis denotes the detected intensity for the red, green and blue channels of the two cameras (at x-axis equal to 0°, the parallel camera 1 camera sees maximun light and the perpendicular camera 2 camera sees zero light).. 29

2.5 - Monte Carlo model and experimental results of a 633-nm wavelength laser beam scattering from a solution of 2-mm microspheres in water. On the left is the image of the experimental polarized image, and on the right is the corresponding image of the Monte Carlo model.. 30

2.6 Error analysis of the optical components of the polarized light camera. 31

2.7 Polarized image of a Melanoma in situ. The image in the top raw are the raw par, per and pol images. The images in the bottom raw are the images after the error calculation.. 33

2.8 Error due to the pixel to pixel variation.. 35

2.9 Error due to the pixel to pixel variation.. 35

2.10 Total error in the polarized light image. ..…..36

2.11 Normal image of melanoma *in situ*. Each pixel in the image was normalized by an average value of the image of a 65% reflectance standard, then multiplied by 0.65. White squares denote two regions of interest: non-pigmented/amelanotic (N) and melanotic (M), from which data for Fig. 2.7 is collected.....33

2.12 Polarized image of melanoma. White squares denote regions of interest, nonpigmented/amelanotic (N) and melanoma (M), from which data for Fig. 2.8 is collected.. 38

2.13 Plot of polarized light image values versus normal reflectance image values for two locations, one with visible melanin M and one with non-pigmented regions N. The value R was (measure of skin / measure of standard) 0.65. The standard was an epoxy block with 65% reflectance across the visible spectrum. The light source is a white LED and the reflectance of the skin and absorption by melanin are wavelength dependent, so the normalization by the standard gives a composite reflectance over a wavelength range. 40

2.14 Image of a nevus. The apparent hilly structure of the nevus is visible in the polarized image on the left. The normal image on the right does not show this structure.. 40

2.15 Polarized (left) and normal (right) image of a squamous cell carcinoma on the upper lip.....41

2.16 Image of a melanoma on a patient nose. The melanoma appears dark in the normal image (right) and white in the polarized image. The arrows show branches of the spreading melanoma. This entire area was excised during surgery.. 37

3.1 Flow chart of Polarized Monte Carlo program, the white cells are used in Standard Monte Carlo programs the gray cells are specific of Polarized Light Monte Carlo programs.....46

3.2 Meridian planes geometry. The photon's direction of propagation before and after scattering is respectively I_1 and I_2. The plane COA defines the meridian plane before and after scattering. 47

3.3 Visual description of the rotations necessary to transfer the reference frame from one meridian plane to the next. Initially the electrical field **E** is defined with respect to the meridian plane COA. 49

3.4 For a scattering event to occur the Stokes must be referenced with respect to the scattering plane BOA. The electrical field **E** (blue lines) is referenced respect to the BOA plane s.. 50

3.5 After a scattering event the Stokes vector is still defined respect to the plane COA. A new rotation of the reference frame is necessary to bring the reference plane back to a meridian plane COB. 51

3.6 Second reference plane rotation. The reference frame is a meridian plane COB.. 52

3.7 The triplet **w, v, u** is rotated of an angle ϕ. The rotation is about the axis **u**, left image, the rotated **w** and **v** vectors are shown on the right. **u** remains unchanged.. 57

3.8 Second rotation on the **w, v, u** triplet. The rotation is of an angle α about the vector **v** as shown on the left. The rotated vector are shown on the right. **v** remains unchanged... 58

3.9 p Rotation about an axis u of an angle ε. This rotation will bring v parallel to the Z axis and the plane w0u in a meridian plane. 59

3.10 Effect of a rotation about an axis **u** of an angle ε on the **w** and **v** vectors. **v** is parallel to the Z axis and the plane **w0u** is in a meridian plane 60

3.11 Left figure shows the Mie phase function for incident linear polarized light (Stokes vector [1,1,0,0]) and the right figure is the phase function for incident circular polarized light (Stokes vector [1,0,0,1]). The sphere diameter is 0.01μm, the incident wavelength is 543 nm, the index of refraction of the sphere is 1.59 65

3.12- Experimental setup, top view. Polarized images of light scattered by a microspheres solution were acquired with a digital camera. Laser light (632.8nm) was polarized and the scattered light was passed through an analyzer whose orientation was either parallel or perpendicular to the incident beam.. 69

3.13 Top image is the experimental results obtained with polarizer and analyzer oriented parallel to each other, the bottom image is the Monte Carlo simulation.. 70

3.14 Comparison of excitance for experiment (green) and Monte Carlo results (black dashed) for parallel image. This intensity profile corresponds to the centerline of the axis of irradiation 71

3.15 Top image is the experimental results obtained with the polarizer oriented parallel to the optical table and analyzer oriented perpendicular to the optical table, the bottom image is the corresponding Monte Carlo simulation. Both images were normalized by the maximum value of the parallel image. 72

3.16 – Comparison of Mueller matrix results for light backscattered from solutions of microspheres. The image on the left is an experimental results of Cameron *et. al.* the image on the Mueller matrix obtained with these Monte Carlo programs. 73

3.17 Experimental apparatus. I studied backscattering from a solution of microspheres (diameter 2μm) when the polarized illumination is at an angle q. Three different light source polarizer orientations (H V P) were used as well as three detectors polarizer orientation (H V P). Images were combined to produce 9 of the 16 element of a Mueller matrix. 74

3.18 First 9 of the 16 element of Mueller matrix for a microsphere solution. The spheres diameter was 2μm and index of refraction 1.59. The Spheres were in a water media with index of refraction equal to 1.33. Each image is 4x4 cm. Light was incident at an angle θ from the normal. 75

3.19 Monte Carlo simulation of the first 9 of elements of the 16 element of Mueller matrix. Illumination was asymmetrical 76

4.1 Simple transmission mode set-up that we can model with the heuristic model. The analyzer can be rotated in two positions: co-polarized and cross-polarized. 81

4.2 Transmission experiment. The second polarizer between the cuvette and detector is an analyzer to select transmitted light either parallel or perpendicular to the axis of the first polarizer. 85

4.3 Experimental data for polystyrene microspheres in water. Sphere diameter was 482nm, μ_s=8cm^{-1}. Wavelength was 543nm, Solid lines are best fit of equation 4.8 and 4.9 (below) with f = 0.005, χ = 0.8, μ_{att} = 1.5 $\beta = 10^{-4}$. 87

4.4 – Experimental results for chicken liver. Lines are best fit to data with f = 0.65 10^{-4}, μ_{att} = 0.45 cm^{-1}, and χ = 0.0007 $\beta = 10^{-5}$... 88

4.5 Summary of diffusivity χ [rad^2/mfp] of polystyrene microspheres and various *ex vivo* tissues. (Left) Diffusivity plotted versus microsphere diameter. (Right) Tissue diffusivity. Data from this study are represented as circles and diamonds. Data from Jarry *et al.* [17] are represented as squares..... 89

4.6 Experimental results and Monte Carlo simulations (black circles and squares) for a mono-disperse solution of microspheres. Diameter of microspheres was 482 nm, light wavelength was 543 nm. The back line is the heuristic model applied to the Monte Carlo data χ= 0.1. The dashed line is the original prediction based only on the experimental data χ=0.07, f=3•10^{-4}, μ_{att}=1.0. The experimental extinction of the polarizer analyzer pair was 10^{-5}..... 91

4.7 Experimental results and Monte Carlo simulations (black circles and squares) for a mono-disperse solution of microspheres. Diameter was 0.308 μm, light wavelength was 0.543 μm..... 91

4.8 Pencil beam (λ= 543 nm) impinging on a slab of variable optical thickness τ, a detector behind an aperture of radius a collects a fraction of the transmitted light. Three apertures of increasing diameter d are studied 0.2 cm, 2 cm and 4 cm, the detector is located at a distance R from the slab. 92

4.9 Monte Carlo simulations (circles) and heuristic model fits for the slab geometry of figure 4.8. Parallel and perpendicular results are modeled with the heuristic model. Model fits parameters are given in table 4.1 93

4.10 The heuristic model fits much better when the μ_{att} parameter is included. Fits results are tabulated in Table 94

4.10 Rotation angle ϕ for traveling through a mono-disperse solution of microspheres of 1μm diameter. The distribution of the angle ϕ is plotted for 1, 6, 8, and 10 scattering event..... 96

4.12 The Poincaré sphere. The coordinates of M are the last three elements of the Stokes vector, [Q, U, V]. The radius of the sphere is the first element of the Stokes vector I..... 97

4.13Ellipticity for 4 sphere sizes (0.01 μm, 0.3μm 2μm and 10μm) the red line is the distribution of ellipticity for 2 scattering events, the green line is the ellipticity of 19 scattering events, the lines in between are the ellipticity distribution fro 3 to 18 scattering events. 99

4.14 Azimuthal angle frequency distribution for 4 sphere sizes. The green line is the distribution of azimuthal angle for 2 scattering events, the red line is the azimuth angle after 19 scattering even. 100

4.15 Representation of the states of polarization in the complex plane.....101

4.16 Complex representation of the status of polarization of the photon after 2 and 19 scattering events. Green data is for 2 scattering events red data is 19 scattering events. 102

4.17 – Geometry of Monte Carlo runs. Two-dimensional maps of polarized light scattering are generated for increasing. 104

4.18 – 10nm spheres. XZ and YZ sections are showed in all graphs. The axis are in cm. Parallel (top image), the crossed transmission (middel image), and the degree of linear polarization of the transmitted light (bottom image). 105

4.19 Transmission through various thickness z, showing x-y variation from 482nm. μ_s=11 cm^{-1}.....106

4.20 Transmission through various thickness z, showing x-y variation from 2000nm. μ_s=11 cm^{-}.....106

4.21 Co-polarized transmission for three sphere sizes.....107

4.22 Cross-polarized transmission for three sphere sizes. 108

4.23 Iso-polarization when POL = .5 for 10nm spheres.....109

4.24 Iso-polarization for POL = 0.9 for 482nm spheres.....109

4.25 Iso-polarization for POL = 0.9 for 200nm sphere.....110

4.26 Three Iso-polarizations for 0.482μm spheres. Three iso-polarizations are shown: in green 0.9 in red 0.6 in blue 0.3. On the top view is visible how the scattering is not azimuthally independent. A slight cross (Maltese cross) is visible.....112

4.27 Parallel and perpendicular polarized transmission for slabs of increasing optical thickness. The sphere sizes considered are: 0.308, 0.482, 0.806, 1.069, and 2.03μm. The parallel polarized light is in red, the perpendicular polarized light is in blue. Notice that for the largest sphere size 2030nm, the parallel polarization decreases faster than for smaller spheres such as 1069.

The anisotropy for 2030nm spheres is 0.8752 for 1069nm spheres is 0.9278. λ=543nm $n_{spheres}$ =1.59 and n_{media}=1.33. 113

4.28 Degree of polarization for a beam of linearly polarized light passing through slabs of optical thickness τ. The degree of polarization for the largest sphere size 2030nm decreases faster than the degree of polarization for smaller sphere such as 1069. 114

4.29 Anisotropy versus spheres diameter for spheres ranging from 0 to 10 μm in diameter. As pointed out in the image the anisotropy for 2030nm (g=0.8752) is smaller than the anisotropy for 1069nm (g=0.9278) and 806nm (g=0.91532)....115

4.30 χ results for fitting of graphs of figure 4.28, the value χ is strongly related to the g value 116

4.31 - Monte Carlo transmission (circles) and equation 4.15 (lines) for solutions of 10nm microsphere 118

4.32 New heuristic model of polarized light transfer through scattering slabs. Monte Carlo runs were conducted for 4 sphere sizes (0.01μm, 0.3μm, 1.0μm, and 2.0μm). χ=0.45 for 0.01 and χ=0.8 for 0.3 μm spheres, χ=0.15 for 1μm, χ =0.2 for 2μm, and, χ =0.1 119

4.33 - New heuristic model of polarized light transfer through scattering slabs. Monte Carlo runs were conducted for 5 sphere sizes Top figure is the parallel polarized transmission the bottom figure is the perpendicular polarized transmissio 120

4.34 An aperture a is at a distance R from a slab. 121

4.35 Monte Carlo simulations and heuristic model for the slab geometry of figure 4.8. χ is kept constant at 0.45. Which shows that c is sensitive to sphere size but not the solid angle of collection of the detector 122

4.36 Results of the heuristic model fit for 8 particle sizes. In the top figure the values of χ related to the particle size is plotted, in the bottom image the results for χ related to the anisotropy parameter g is reported 123

5.1 Experimental apparatus. Illumination light was linearly polarized parallel to the scattering plane that corresponds to the source/tissue/camera triangle. The analyzer was a linearly polarizer oriented first parallel then perpendicular to the scattering plane yielding two images, denoted PAR and PER,

respectively. A rough metal band was placed on a thick tissue slab and an overlaying thin tissue section of thickness L was placed over the metal band. 127

5.2 Images of Pol = (PAR-PER)/(PAR+PER). Liver at 696 nm wavelength. Left column shows PER images, and right column shows POL images. (A) Metal band with no overlaying tissue (L = 0). (B) Metal band covered on left by overlaying tissue of thickness L = 420 μm. Also shown are two rectangular regions of interest, one over metal and one not over metal. (C) Metal band covered on left by overlaying tissue of thickness L = 1020 μm 130

5.3 Example of experimental data. The tissue was liver. The values of D = (PAR-PER)metal – (PAR-PER)tissue versus thickness L of the tissue that overlays the metal band. Data for four wavelengths (480, 543, 633, and 696 nm) shown. Data at zero thickness is low because incident light is obliquely reflected directly off metal and misses the camera.. 131

5.4 Probability density functions for estimate of 1/e depth of imaging, $p(z_e)$. The choice of z_e was varied from 0 to 600 μm and the prediction of $D = (par - per)_{metal} - (par - per)_{tissue}$ was calculated. A function $p(z_e)$ was calculated equal to 1/error where error equaled the sum of squared differences between experimental data and predicted values for D. The $p(z_e)$ was normalized to 1 at its peak value to aid comparisons, consequently the units of $p(z_e)$ are in arbitrary units. The three graphs show the skin, muscle and liver curves at each of the four wavelengths tested 132

5.5 The $1/e^2$ depth of imaging ($= 2z_e$) as a function of wavelength for three tissues (skin, muscle, liver). The values are the values of 2ze corresponding to the peaks of Fig. 5.4. The error bars indicate the full-width half-max of the distribution for the estimate of $2z_e$. 133

5.6 Left column is L = 400μm. Right colume L = 1200μm. (A)PER. (B) PAR-PER. (C) POL..... 135

5.7 Contrast for polarized and unpolarized images. The blue curve obtained comparing areas of background tissue and area on the rough metal band for PER images, the red curve was obtained comparing the same areas in PAR-PER images. The contrast for PAR-PER images is more that twice as big as the PER images contrast. Polarized light imaging aids in situations where the depth of imaging is shallow and enhances contrast in the superficial layers of the skin.. 136

5.8 Experimental apparatus. 139

5.9. Train of polarizing elements used for Mueller matrix type experiments. LP stands for linear polarizer QW for quarter-wave plate. The orientation of the polarizing elements is described in table 5.....141

5.10 Alternative train of polarizing elements used for Mueller matrix type experiments. LP stands for linear polarizer QW for quarter-wave plate and HW for half-wave plate. The orientation of the polarizing elements is described in table 5.....141

5.11 Data matrix (DM). Each element corresponds to the intensity of transmitted collimated light when the source was either H, or V, or P or R and the detector was either H, or V, or P or R. 142

5.12 Reconstructed Mueller matrix, all the elements are obtained from table 5.2.....145

5.13 (Left) Ideal Mueller matrix of air. (Center) Experimental Mueller matrix for air (Right) Corrected Mueller Matrix of air which accounts for the non-ideality of the optical components of the system. 147

5.14 First example, a system is composed of a detector (D) two linear polarizer (LP) and source (S.. 149

5.15 Uncertainty in an HH measurement, the only affected component is the U component. In the range of 1 the uncertainty is less than 2. 150

5.16 Transmission versus tissue thickness. (Top) Skin (porcine). (Middle) Muscle (chicken breast). (Bottom) Liver (chicken).. 152

5.17 Data Matrix for 5 positions (y = 1 to 5) on 3 skin samples of thickness 170, 240 and 370 mm (x = 1 to 3, respectively). Color map = $\log_{10}(T)$. 153

5.18 Mueller matrix for skin sample 154

5.19 Data Matrix for 5 positions (y = 1 to 4) on 4 muscle samples of thickness 150, 270, 360 and 420 mm (x = 1 to 4, respectively). Color map = $\log_{10}(T)$. 154

5.20 Mueller matrix for chicken muscle sample 155

5.21 Data Matrix for 5 positions (y = 1 to 5) on 4 liver samples of thickness 200, 360, 420 and 900 mm (x = 1 to 4, respectively). Color map = $\log_{10}(T)$. 156

5.22 Mueller matrix for liver sample.....156

5.23 Skin. (TOP) Data Matrix values versus tissue thickness L (μm). (BOTTOM Mueller Matrix, normalized by MM(1,1), versus L 158

5.24 Muscle. (TOP) Data Matrix values versus tissue thickness L (mm). (BOTTOM Mueller Matrix, normalized by MM(1,1), versus L. 159

5.25 Liver. (TOP) Data Matrix values versus tissue thickness L (mm). (BOTTOM Mueller Matrix, normalized by MM(1,1), versus L 160

5.26 The loss of Total Degree of Polarization versus tissue thickness (L [mm]) through which the laser beam passes. Tissues are skin, muscle and liver. 162

5.27 Schematic of the polarized microscope.....164

5.28 The ten corresponding pixels in ten transmission images are analyzed versus angle between source and analyzer polarizer. 165

5.29 The transmission versus angle data was fit as $A\sin^2(\theta) + B\cos^2(\theta)$ + offset versus θ. The shift Δθ was determined.....165

5.30 Muscle sample. (Top) Transmission image for θ =90 (crossed polarizers) . (Bottom). Map of the retardance, Δθ [degrees]. The size of iso-retardance domains is on the order of 10-100 μm 166

6.1 Front view of the goniometric assembly. The optical fiber rotates around a pivotal point; the fiber is mounted on a Delrin arm. The exact angular position of the incident beam is measured with a protractor.....172

6.2 Detail of the fiber support arm. The laser spot is centered on the pivotal plane for every θ angle of the support arm. The sample is maintained at the center of the pivotal plane with glass supports. A pedestal connected to the microscope table with double-sided tape assures the rigidity of the system.....173

6.3 Collection model showing the definition of angles used. The numerical aperture of the objective lens is defined as N.A. = n sin (θ). The direction of propagation is in the positive z-direction.. 175

6.4 Modeled behavior of perpendicular polarized incident light and perpendicular collection by a microscope objective lens for various numerical aperture for spheres of diameter = 2.03 μm. The plotted numerical apertures are 0.0548, 0.1,0.3, and 0.55. The spheres had an index of refraction of 1.59 and were in a medium with index of refraction 1.33 176

6.5 Perpendicular polarized light scattered by a sphere of diameter 6.4 μm and a sphere of diameter 1.03 μm. The N.A. of the objective lens was 0.56. Scaling factor f_s = 2.5 for the 6 μm sphere and 1.8 for the 1 μm sphere... 177

6.6 Parallel polarized light scattered by one sphere of diameter 2.03 μm. Numerical aperture is 0.095. Scaling factor f_s = 0.1. 178

6.7 Perpendicular polarized light scattered by one sphere of diameter 2.03 μm. Aperture 230 μm and N.A.=0.055. Scaling factor fs = 2.1. 178

6.8 Perpendicular polarized light scattered one by sphere of diameter 6.4 μm. Aperture 230 μm and N.A.=0.055. Scaling factor fs = 1.1.. 179

6.9 Total error in total scattering for 2.03 μm diameter microspheres. The size with the minimum used to establish the sphere size. In this case 2.11 μm. Two other minima are visible in the figure; this could constitute a problem when trying to determine the particle size.....180

6.10 - Modeled behavior of perpendicular polarized light scattering from a modeled cell. The relative index of refraction was 1.036. The nucleus size varied from 2 to 15 microns. Because epithelial cell nuclei range from 5 to 10 μm in diameter, I can expect to be able to detect nucleus enlargement and scattering from smaller particulates such as organelles.. 181

6.11 Images of polarized light scattering from muscle cells when incident light is at 30˚ respect to the sample normal. Nucleus and cytoplasm are visible for the larger cell stretching across the image. This image is very bright with intensity levels around 50,000 counts.....182

6.12 Images of polarized light scattering from muscle cells when incident light is at 40˚ respect to the sample normal. Some structures inside the cell appear brighter than the surrounding area. In this image the intensity has dropped to 9000 counts, less and less direct light is entering the aperture of the objective lens.....182

6.13 Images of polarized light scattering from muscle cells when incident light is at 50˚ respect to the sample normal. Some structures inside the cell appear brighter than the surrounding area. In this image the intensity level dropped to less than 1000, only light whose original direction was deviated by scattering enters the objective lens... 183

6.14 Graph of scattered intensity versus the illumination angle for pixels within the nucleus and the cytoplasm of the triangularly shaped muscle cell. Cytoplasm scatters light more broadly than the nucleus.. 183

7.1 Skin can be modeled as a spheroids of different sizes, a Monte Carlo program will be written to describe the effect on polarization of these layers.....190

Abstract

Imaging skin pathologies with polarized light: empirical and theoretical studies

Jessica C. Ramella-Roman
OGI School of Science and Engineering
Oregon Health and Science University

Supervising Professor: Steven L. Jacques

The use of polarized light imaging can facilitate the determination of skin cancer borders before a Mohs surgery procedure. Linearly polarized light that illuminates the skin is backscattered by superficial layers where cancer often arises and is randomized by the collagen fibers. The superficially backscattered light can be distinguished from the diffused reflected light using a detector analyzer that is sequentially oriented parallel and perpendicular to the source polarization.

A polarize image pol = parallel-perpendicular / parallel+perpendicular is generated. This image has an higher contrast to the superficial skin layer than simple total reflectance images. Pilot clinical trials were conducted with a small hand-held device for the accumulation of a library of lesions to establish the efficacy of polarized

light imaging in vivo. It was found that melanoma exhibits a high contrast to polarized light imaging as well as basal and sclerosing cell carcinoma. Mechanisms of polarized light scattering from different tissues and tissue phantoms were studied in vitro. Parameters such as depth of depolarization (DOD), retardance, and birefringence were studied in theory and experimentally. Polarized light traveling through different tissues (skin, muscle, and liver) depolarized after a few hundred microns. Highly birefringent materials such as skin (DOD = 300 μm @ 696nm) and muscle (DOD = 370 μm @ 696nm) depolarized light faster than less birefringent materials such as liver (DOD = 700 μm @ 696nm). Light depolarization can also be attributed to scattering. Three Monte Carlo programs for modeling polarized light transfer into scattering media were implemented to evaluate these mechanisms. Simulations conducted with the Monte Carlo programs showed that small diameter spheres have different mechanisms of depolarization than larger ones. The models also showed that the anisotropy parameter g strongly influences the depolarization mechanism. Large spheres will depolarize faster than smaller sphere if their anisotropy is smaller. A linearly polarized beam traveling through a solution of 1.07 μm microsphere (g = 0.9278 @ 543 nm) depolarizes slower than in a solution of 2.03 μm (g = 0.8752 @543 nm) microspheres . The Monte Carlo programs were also used to test two heuristic models of polarized light transport into scattering media. The models were based on a heuristic parameter χ [radians^2/mfp], based on the apparent diffusivity of the angle of orientation for linearly polarized light. Large χ corresponded to highly depolarizing media, and small χ correspond to low depolarizing media. A high χ parameter was found for solutions of small spheres, and for highly birefringent biological media. Larger spheres showed lower values of χ. A novel asymmetric illumination microscope was implemented to investigate precancerous nuclei enlargement. Preliminary experiments were conducted on latex microspheres, the diameter of the sphere could be obtained with only *a priori* knowledge of the sphere index of refraction mismatch.

Chapter 1
Introduction

Defining the true border of skin cancer reliably and non-invasively is still an unsolved problem for the dermatology community. In this dissertation a new non-invasive method to visualize cancer borders using polarized light imaging is discussed. This technology will assist dermathologists in making a quick assessment of skin cancer borders. The basic concept of a polarized light video system is schematically illustrated in Figure 1.1. Incident light is linearly polarized, oriented parallel to the scattering plane defined by the source-tissue-camera plane. This light illuminates the skin obliquely at about 20° off the normal. A glass plate contacts the skin in the image plane to ensure a flat surface. Glare due to specular reflectance off the air/glass interface and the glass/skin interface is deflected obliquely and misses the camera. The camera views the skin perpendicular to the surface. Only light scattered from within the skin passes through the analyzer (linear polarizer) before being imaged by the camera.

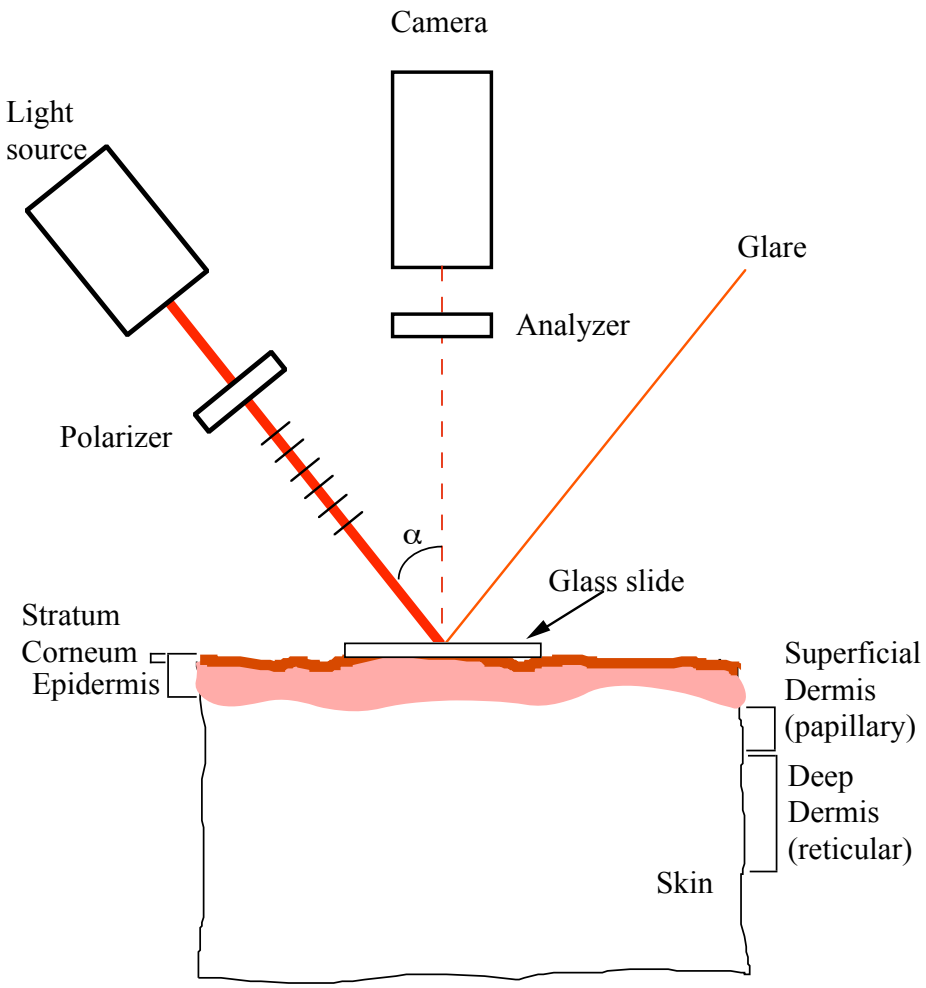

Fig. 1.1 – Simple schematic of polarized light imaging. A light source is oriented at an angle to the skin. The light beam is polarized parallel to the source/sample/detector plane. Some of the light penetrating the skin layers is backscattered in the detector direction. This light is analyzed with an analyzer oriented either parallel (PAR) or perpendicularly (PER) to the source polarization. A polarized image is reconstructed with the two PAR and PER images using equation 1.1.

Two images are acquired: (PAR) the first image has the analyzer aligned parallel with the scattering plane just like the source, (PER) the second image has the analyzer aligned perpendicular to the scattering plane. The images are comprised of backscattered photons from superficial layers that largely retain the polarization orientation of the source. These photons contribute more strongly to the PAR image. Light returning from deeper layers is multiply scattered and these photons are equally likely to be found in the

parallel or perpendicular polarization state. Consequentely, these photons will make equal contributions to PAR and PER.

Thus

$$PAR = (\ Superficially\ scattered\ light + Deeply\ scattered\ light)$$

$$PER = Deeply\ scattered\ light$$

The difference image is calculated to reject the deeply scattered light:

$$PAR - PER = Superficially\ scattered\ light$$

The variation in the illumination (I_o), and the superficial melanin pigment (T_{mel}) effects equally PAR and PER.

$$PAR = I_o T_{mel}(\ Superficially\ scattered\ light + Deeply\ scattered\ light)$$

$$PER = I_o T_{mel}\ \ Deeply\ scattered\ light$$

Normalization of the difference image by the sum image cancels both these effects:

$$POL = \frac{PAR - PER}{PAR + PER} \tag{1.1}$$

The polarized image focuses only on the superficial layer of the skin eliminating any artifact due to superficial pigmentation and source disuniformity.

$$POL = \frac{(I_o T_{mel} Superficial + I_o T_{mel} Deep) - I_o T_{mel} Deep}{(I_o T_{mel} Superficial + I_o T_{mel} Deep) + I_o T_{mel} Deep} \tag{1.2}$$

$$= \frac{Superficial}{Superficial + 2Deep} = \frac{Superficial}{Total}$$

A clinical prototype was built by Professor Steven L. Jacques and Doctor Ken Lee. An image of the early clinical prototype is shown in Figure 1.2.

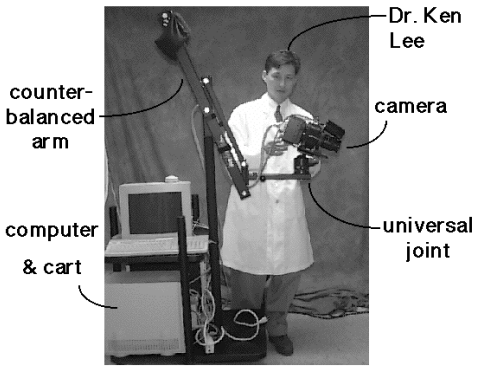

Fig. 1.2 - First clinical prototype tested at OHSU Dermatology, with Dr. Ken Lee

Figure 1.2 shows the first clinical prototype that was used in a pilot clinical trial. The camera was positioned on a universal joint located on a boom. A counterbalance arm was used to stabilize the camera arm. Figures 1.3, 1.4, 1.5 illustrate some of the in vivo images that have been acquired with this polarized light camera.

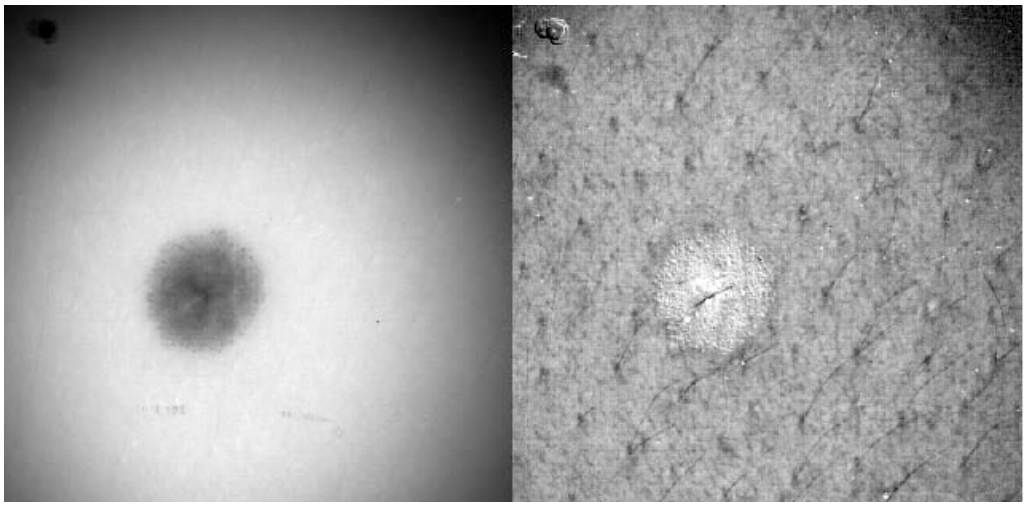

Fig. 1.3 - Image of a benign pigmented nevus. The left-hand side image is unpolarized, and the image on the right is the polarized image. Notice how illumination variation over the image is corrected in the right-hand side image.

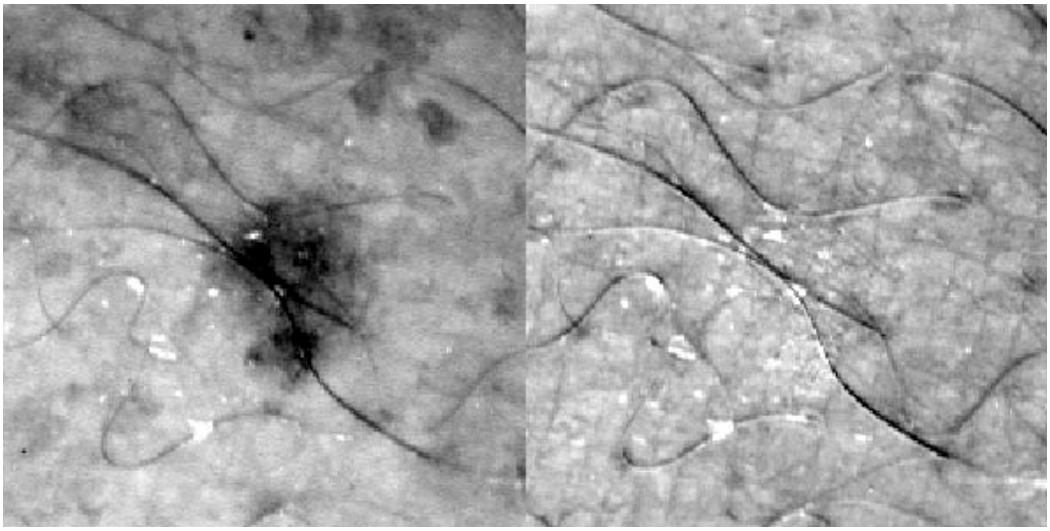

Fig. 1.4 - Image of a freckle. The left-hand side image is unpolarized. The image on the right is the polarized image. The freckle disappears in the polarized image, because superficial melanin absorption is cancelled in the POL equation (1.2).

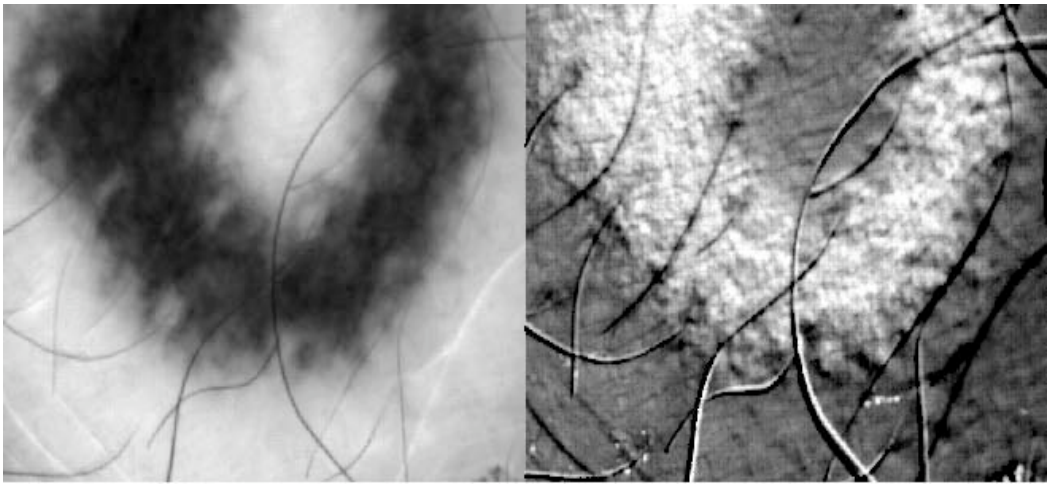

Fig. 1.5 - Image of a tattoo. The left-hand side image is unpolarized. The image on the right is the polarized image. Tattoo pigment are enhanced by the polarized camera.

Figure 1.3 is an image of benign pigmented nevus. The normal image shows only the pigmentation. The polarized image, POL, ignores the pigmentation and shows the granular structure of nevus and other superficial detail; for example notice the epidermal infolding around every hair follicle. Figure 1.4 is an image of freckle. This superficial

melanin pigmentation is canceled in the POL image as in equation 1.2. The third Figure, 1.5 is an image of a tattoo. The normal image shows the black tattoo. The polarized image reveals the specular reflectance off the individual carbon particles.

1.1 Motivation

Half of all new cancers are skin cancers and about 1.4 million new cases of skin cancer are diagnosed in the United States each year [1]. The first detection of skin cancer is usually performed by the patient himself noticing an abnormal or enlarged lesion. A visual inspection by primary care clinicians and dermatologist usually follows. Clinicians often use the "ABCD rule" (Asymmetry, Border irregularity, Color variegation, and Diameter greater than 6 mm) to detect melanoma. This is a widely accepted tool to promote early detection. Doctors check for border irregularity, since skin cancer borders tend to be fragmented with pod-like extensions. Irregular coloring is a flag for most skin cancers: some cancers have shades of brown, black, blue-gray, pink, and white; nevertheless sometimes skin cancer can be asymptomatic. Lesions whose diameter (D in the ABCD rule) is larger that 6 mm and that are much larger than the patient's other skin anomalies should be tested. Other more sophisticated diagnostic techniques are available to clinicians; an overview of both invasive and non-invasive diagnostic techniques will be given in the next paragraphs.

Accurate detection depends on doctor training; an examiner with 10 or more years experience can successfully detect melanoma 80% of the time versus 60% for less experienced clinicians [2].

1.2 Skin Cancer

There are three main types of skin cancer: squamous and basal cell carcinoma and melanoma. Polarized light imaging of these lesions, some which are shown in Figure 1.6,1.7 and 1.8, showed some interesting features. The images taken by Jacques *et al.* [3], have shown a promising sensitivity in particular for malignant and sclerosing basal

cell carcinoma. The use of polarized light imaging combined with a dermatologist expertise could greatly increase the definition of true borders of skin cancer.

1.1.2 Squamous Cell Carcinoma

This kind of cancer consists of a firm irregular fleshy growth caused most of the time by a sunburn. The growths tend to ulcerate. If untreated, the cancer may spread to the surrounding lymph glands. Squamous Cell Carcinoma is common in elderly patients especially in cases of continuous exposure to the sun over the years. Figure 1.6 shows an example of squamous cell carcinoma taken with the polarized light and without polarization.

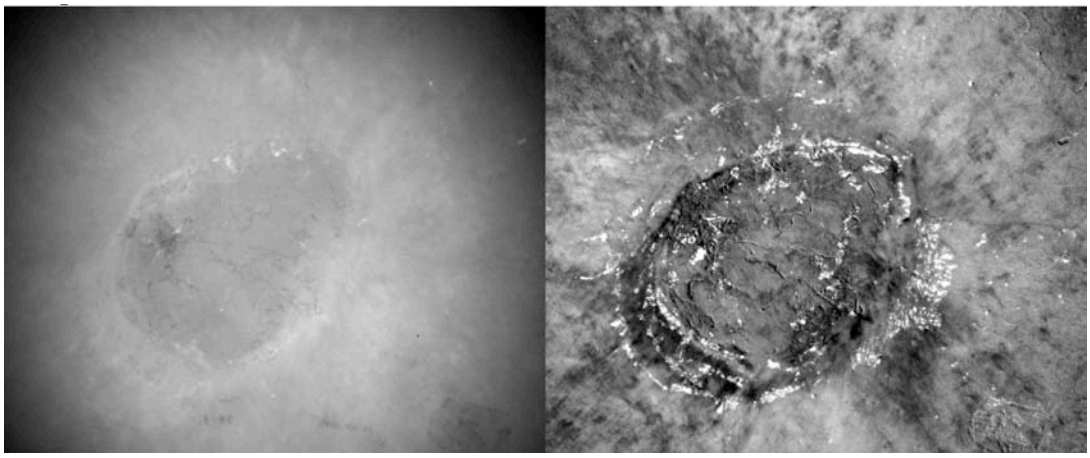

Fig. 1.6 – Left-hand side is the unpolarized light image. The right-hand side is the polarized light image. Images were taken in the clinic with a manual system (fig 1.2) [3] consisting of a 16-bit digital camera and two high quality polarizers. The white areas in the right-hand side image indicate air bubbles under the glass plate in contact with the skin.

1.2.2 Basal Cell Carcinoma

Basal cell carcinoma is the most common form of skin cancer, accounting for about 80 percent of all new cases. It is slow-growing and painless. It often consists of a pigmented ulcer with a shiny or translucent border. This cancer commonly appears on the face.

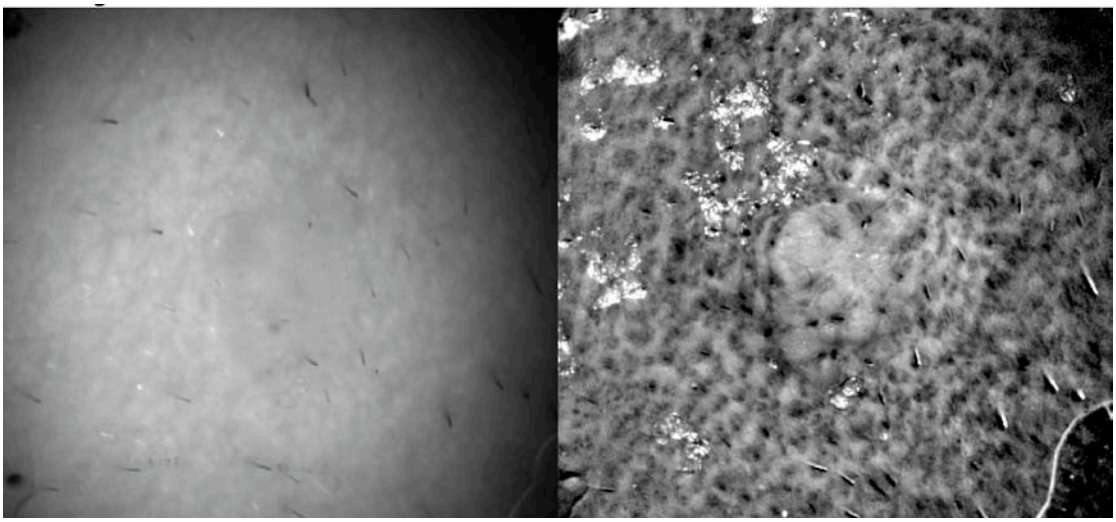

Fig. 1.7 - Basal cell carcinoma. The left image was taken with unpolarized light. The right image was taken with polarized light. The system used for these images was that of Figure 1.2 [3]. The white areas in the right-hand side image indicate air bubbles under the glass plate in contact with the skin.

The image on the left of Figure 1.7 is an unpolarized image of basal cell carcinoma. Only a faint shadow is visible; the image on the right shows the same lesion acquired with a polarized light camera, higher contrast is achieved.

1.2.3 Melanoma

About 41,600 people in the U.S. will develop melanoma this year [1]. This is a cancer of the pigment cells of the skin and is a highly malignant. It is generally dark-colored and appears as a growth but can be easily confused with moles. It grows rapidly with irregular margins and coloring. Melanoma commonly occurs on the fingers, toes and face.

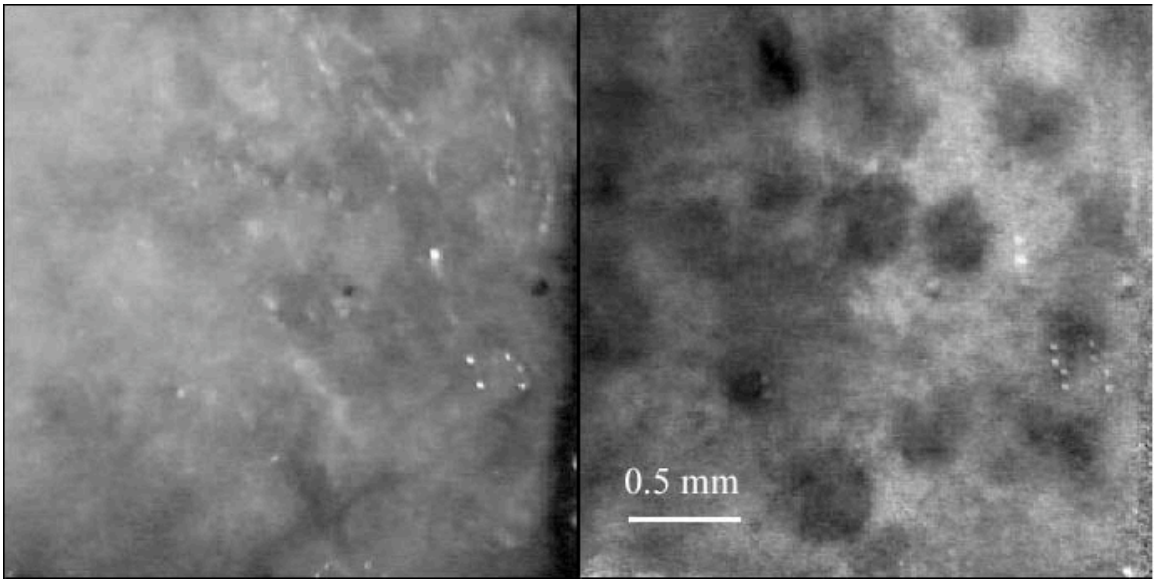

Fig. 1.8 - Image of a melanoma *in situ*. The right-hand side image was taken with a polarized handheld device described in chapter 2. On the left-hand side is the image captured with parallel horizontal detector and source polarizers.

1.3 Non-invasive diagnostic techniques

There are different non-invasive diagnostic methods available to dermatologists, some based on polarimetry, and others based on absorbance spectroscopy. Dermatoscopy (epiluminescence microscopy, dermoscopy, oil immersion microscopy, skin-surface microscopy) is commonly used in the clinic to observe the dermoepidermal junction [4]. Commercially available tools such as Dermlite® photo and Dermlite® platinum (3 Gen, LLC, Dana Point, CA) based on cross-polarization imaging, have yielded good results [5] particularly in the elimination of glare and shadows from the field of view. Systems based on confocal microscopy [6,7,8] allow high resolution imaging of dermis up to 500 μm deep. The drawback of confocal microscopy is the cost and encumbrance Optical Coherence Tomography (OCT) and Polarization Sensitive Optical Coherence Tomography (PS-OCT) apparatus are rapidly moving from the laboratory to the clinic [9]. In PS-OCT the polarization state of the recorded interference fringe intensity is used to enhance the image contrast. High frequency ultrasound has

been used to observe the *stratum corneum* and part of the epidermis [10]. Magnetic resonance imaging has been used for microscopic observation of human skin [11].

The Wood's lamp, a ultraviolet radiation emission lamp (peak wavelength 365nm), is commonly used in dermatology for detection of fungi and several other lesions; this lamp can also be useful in the detection of leukoderma associated with melanoma [12].

Some groups are pursuing spectroscopic methods to enhance the borders of cancerous lesions in particular melanoma [13]. An example is the commercially available device Siascope (Spectrophotometric Intracutaneous Analysis). This device acquires 8 narrow-band images of the lesion from 450 to 950 nm. Using proprietary algorithms, the device calculates the total content of melanin in the epidermis and papillary dermis as well as the collagen and hemoglobin content. This device showed 80% specificity and 82% sensitivity for melanoma discrimination [2].

1.4 Invasive diagnostic techniques

The diagnostic technique that gives the most accurate results in skin cancer discrimination is Mohs surgery. Mohs surgery is performed most often in a hospital. The clinically obvious anomaly is excised under local anesthesia by a dermatologist. The margins of excision are stained and frozen, and then examined under a microscope. If the margins of the excised tissue test positive for cancer, the patient has to undergo further surgery until all margins are negative. This is a laborious, time consuming, and expensive method that can be exhausting to the patient who is often left waiting for a decision from the biopsy laboratory. Another invasive diagnostic technique is the fine needle aspiration for metastatic melanoma [14]. In this technique suspect tissue or fluid is removed with a needle. The removed tissue is then biopsied. The advantage of this technique is its minimal invasivity.

1.5 Polarized light

Polarized light related phenomenons have fascinated scientists for many centuries. The Vikings use of skylight polarization for navigating is a well known albeit unproved

legend [15, 16] dated circa 700 CE. The first documented use of observation of polarized light is attributed to Bertholonius [17]. A Danish mathematician, Bertholonius used a crystal of Iceland spar to discover polarization by double refraction; he also wrote the first report on polarized light [18]. Over the last four centuries we have reached a much greater understanding of the physics behind light transport and polarized light; an interesting and exhaustive historical review on the main discoveries and contributors of in polarization can be found in the first chapter of Christian Brosseau's treatise on statistical optics and polarized light [19].

Polarized light is currently used in many different fields of science, from astronomy and astrophysics [20,21], to oceanography [22,23], microscopy [24], and mechanical testing [25].

1.6 Polarized light in medicine

The use of polarized light in medicine is fairly recent. Bickel *et al*. [26] in 1976 analyzed scattering of polarized light by *Bacillus subtili.* Anderson [27] in 1991 used a method based on cross polarization to analyze some cancerous skin features. His method consisted on accenting or rejecting surface glare by viewing the skin through an analyzer oriented parallel or perpendicular to the incident polarized beam. Schmitt *et al*. [28] reported a study on linear and circular polarized light scattering by turbid media. Jacques *et al*. [29] and Ostermeyer *et al*. [30] have used video reflectometry to study the point spread function of microspheres solutions and skin. Hielscher *et al*. [31] computed the full Mueller matrix from backscattering of highly scattering media and studied the influence of particle size. Maitland [32] studied linear birefringence during heating of native collagen as well as Sankaran [33]. Mourant *et al.* [34] studied the wavelength dependence of polarized light on normal and cancerous cells and their group also developed [35] a fiber optic probe sensitive to polarized light that is effective in the determination of particle size and density of monodisperse solutions of microspheres.

Jarry *et al.* [36] did transmission measurements in tissues. Jacques *et al.* [37] suggested a simple heuristic model of polarized light transfer into scattering media.

Beckman *et al.,* Perelman *et al.* [38, 39], and Sokolov *et al.* [40] have studied the wavelength dependence of polarized light backscattering from nuclei and epithelial cell to establish their size and index of refraction. Several groups [20,41,42,43,44,45,46] have compiled Monte Carlo programs based on polarized light scattering from small spheres (in the Mie or Rayleigh regime). Wang *et al.* [47] extended their Monte Carlo program to birefringent turbid media and showed the first theoretical images of polarized light backscattered from such media.

1.7 What is polarized light?*

There are many books describing the fundamentals of polarized light: Born and Wolf's [48] rigorous book on optics, Brosseau's [19] volume on statistical optics, Clark and Clarke and Grainger's [49] book on polarized measurements, and Huard's [49] extensive treatise to cite a few. A great introductory book was written by Shurcliff and Ballard [18].

A complex phenomenon such as a beam of white light can be greatly simplified by passing it through a prism. The various wavelengths composing the white light beam will spread into beams of different wavelengths that can be considered separately. If now we pass one of these single wavelength beams through a polarizer we can obtain the simplest kind of light: polarized monochromatic light.

Light can be described as an electromagnetic wave phenomenon. The electric field **E** of an electromagnetic wave can be described as the vector sum of two electrical field components, called $E_{\|}$ and $E_{\perp}$, that are perpendicular to each other. The relative magnitude and phase of these $E_{\|}$ and $E_{\perp}$ electric fields specify the polarization status of the photon (Fig. 1.9). The x, y and z axes of Figure 1.9 are consistent with the right-hand rule specifying that propagation is along the positive z-axis. How the total electric field **E** is divided into $E_{\|}$ and $E_{\perp}$ depends on the frame of reference used to define x and y.

---

*Parts of this paragraph were published in a book chapter: Jacques S.L. and Ramella-Roman J.C. "Polarized light imaging", Lasers and current optical techniques in biology, ESP book series "Comprehensive series in Photo-sciences" D.P. Heater G. Jori (in print)

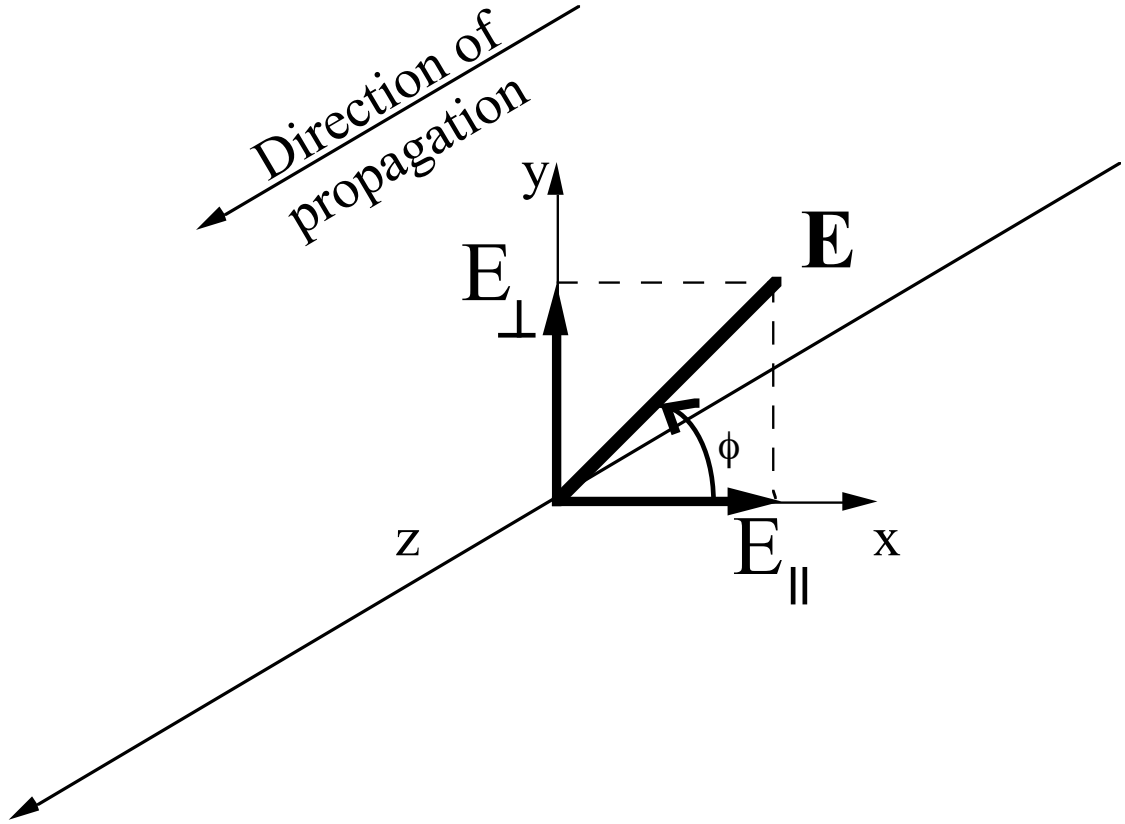

Fig. 1.9 - The $E_{\parallel}$ and $E_{\perp}$ components of the total electric field **E** are shown aligned with the x and y axes, respectively. The angle ϕ indicates the orientation of the E field at a moment in time, viewed as the wave approaches the observer. The wave is propagating along the z-axis toward the lower left toward the observer.

While the magnitude and phase of $E_{\parallel}$ and $E_{\perp}$ can take on any values yielding a continuum of possible states, there are 6 types of polarized light that are commonly used to experimentally characterize the polarization state of light (Fig. 1.10). Let $E_{\parallel}$ be aligned parallel to the x-axis in the horizontal surface (x-z plane) of our experimental table. Let $E_{\perp}$ be aligned perpendicular to the surface of our experimental table. Then the 6 types of polarized light are:

H: The vertical wave ($E_{\perp}$) component has zero magnitude, and the total wave is horizontal linearly polarized ($\phi = 0°$).

V: The horizontal wave component ($E_{\parallel}$) has zero magnitude, and the total wave is vertical linearly polarized ($\phi = 90°$).

P^+: The two components $E_{\parallel}$ and $E_{\perp}$ are aligned in phase and equal in magnitude, and the sum of the two waves is +45° linearly polarized ($\phi = 45°$).

P^-: The two waves are 180° out of phase but equal in magnitude, and the sum of the two waves -45° linearly polarized ($\phi = -45°$).

R: The two wave components are equal in magnitude but $E_{\perp}$ leads $E_{\parallel}$ by 90° in phase, and the sum of the two waves is a right circularly polarized (ϕ rotates counterclockwise as the photon approaches the observer).

L: The two wave components are equal in magnitude but the $E_{\perp}$ lags $E_{\parallel}$ by 90° in phase, and the sum of the two waves is left circularly polarized (ϕ rotates clockwise as the photon approaches the observer).

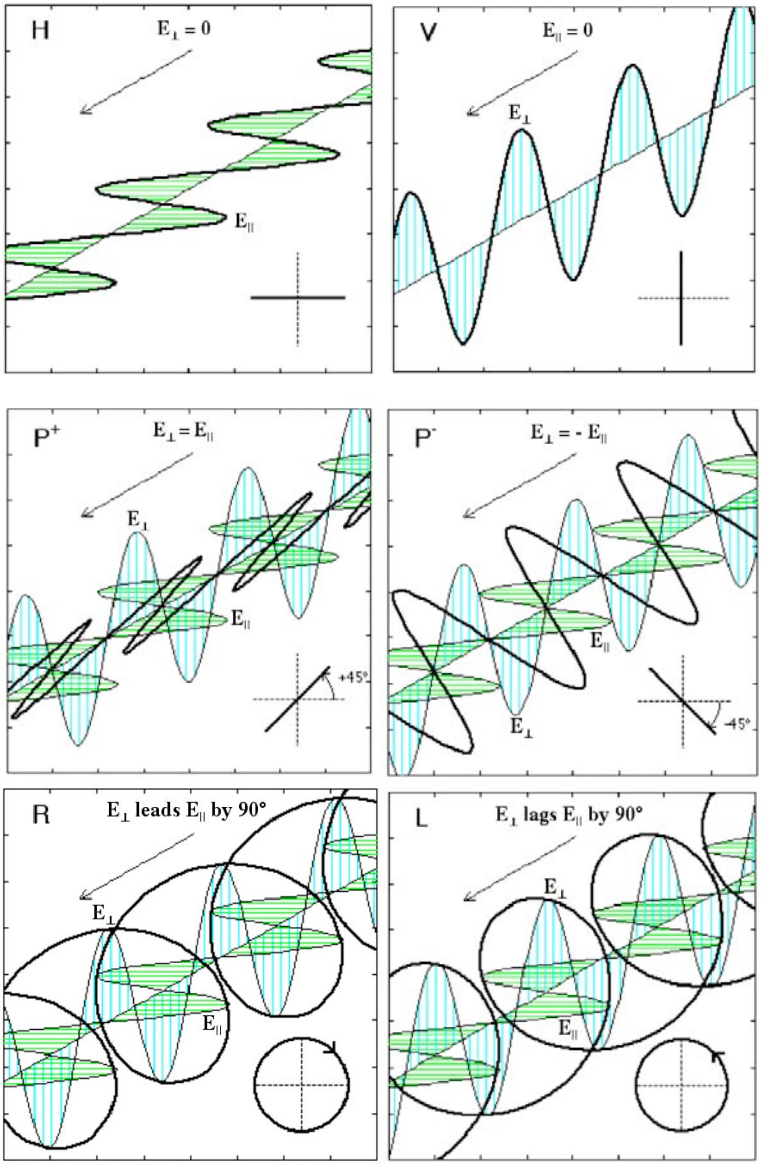

Fig. 1.10: The electric fields for linear H, V, P+ and P- polarized light and for circular R and L polarized light.

1.8The Stokes vector and Mueller matrix formalism

G. G. Stokes introduced the 4 dimensional vector representation of polarized light in 1852 [51]. The Stokes parameters have been named in many different ways. Throughout this thesis the four components are I, Q, U and V. I is the intensity of the beam, Q is the balance between parallel or perpendicular linearly polarized intensity. U is the balance

between plus or minus 45˚ polarized intensity. Finally V is the balance between right and left circular polarization intensity.

More rigorously, using the expression for quasi-monochromatic light:

$$\boldsymbol{E}(t) = \boldsymbol{E}_x(t) + \boldsymbol{E}_y(t) \tag{1.3}$$

where

$$\begin{aligned} \boldsymbol{E}_x(t) &= \boldsymbol{i}\, E_{ox}(t)\cos[(kz - wt) + \varepsilon_x(t)] \\ \boldsymbol{E}_y(t) &= \boldsymbol{j}\, E_{oy}(t)\cos[(kz - wt) + \varepsilon_y(t)] \end{aligned} \tag{1.4}$$

The Stokes vector is:

$$\begin{aligned} I &= \left\langle E_{ox}^2 \right\rangle + \left\langle E_{oy}^2 \right\rangle \\ Q &= \left\langle E_{ox}^2 \right\rangle - \left\langle E_{oy}^2 \right\rangle \\ U &= \left\langle 2E_{ox}E_{oy}\cos\varepsilon \right\rangle \\ V &= \left\langle 2E_{ox}E_{oy}\sin\varepsilon \right\rangle \end{aligned} \tag{1.5}$$

Where the angle brackets indicate time-averaged values, ε is the phase angle, and E_{ox} and E_{oy} are the horizontal and vertical components of the instantaneous electric fields.

The success of the Stokes representation over the centuries is due in part to its ability to predict how a polarized beam will be affected by polarizing elements such as polarizers and waveplates. Moreover Stokes vectors can be added giving an idea of the total polarizations state at a point in space. These properties of the Stokes vectors were used in the development of three Monte Carlo programs to model the status of polarization of a beam traveling through scattering media.

The Stokes vectors are generally used in combinations with Mueller matrices. Mueller matrices are mathematical representations of polarizing elements. They are four-by-four matrices that in combination with Stokes vectors can model a beam passing through various optical elements. Given a Mueller matrix **M** for a particular polarizing element and a beam of light incident on that element represented by a Stokes vector $[I\ Q\ U\ V]_{Input}$ the output Stokes vector can be calculated as:

$$\begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix}_{Output} = \boldsymbol{M} \times \begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix}_{Input}$$

Some Mueller matrices are shown in below.

$$linear\ \ polarizer\ \ at\ \ 0^\circ = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.6}$$

$$linear\ \ polarizer\ \ at\ \ 90^\circ = \frac{1}{2}\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$linear\ \ polarizer\ \ at\ \ +45^\circ = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$linear\ \ polarizer\ \ at\ \ -45^\circ = \frac{1}{2}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$right\ \ circular\ \ polarizer = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$left\ \ circular\ \ polarizer = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

1.9 Thesis goals

The polarized light camera provides doctors with another tool to increase their ability to detect the borders of skin cancer. This tool will help them make a more accurate assessment of the extent of the cancerous lesion. This should minimize patient discomfort by limiting the excision to just the lesion, and should also help the surgeon to exercise lesion infiltrates that might otherwise be overlooked.

The main goal of these studies was to discover the mechanism behind the contrast enhancement obtained with polarized light imaging.

A new polarized light camera was needed in the clinic to acquire a large library of polarized skin lesions. This was necessary to establish which lesions exhibit more contrast to the polarized light camera. Another important goal for the new device was small encumbrance, reduced cost, and the ability to capture images automatically. A prototype was built for few hundreds dollars and was able to capture images with good contrast. This camera was developed as a proof of principle, the developed control software can be used with any standard digital camera.

Chapter 2, describes this small hand-held device and several images that were acquired at the OHSU Dermatology department. Images were acquired minutes before excision of anomalous skin lesions, and the complete set of more than 300 pictures is available on CD. The source code for this device is in Appendix 1.

Polarized light transfer into scattering media such as Rayleigh or Mie particles is a well known phenomenon. However flexible models such as Monte Carlo programs were not available to us. I developed three robust Monte Carlo models of polarized light transfer into simple scattering media, that yield the same results. The programs were used to model experiments conducted with microspheres. Using the Monte Carlo program as a gold standard a better understanding of polarized light travel in biological media was achieved. Moreover, some cell structures can be simulated as a collection of small spheres; the scattering of polarized light from these structures can be modeled with the polarized light Monte Carlo model.

In chapter 3, Monte Carlo programs are described. Extensive testing of these programs is also shown and the source codes can be found in Appendix 2.

Polarized light Monte Carlo programs are very useful, but they are time consuming and do not consider non-spherical particles and birefringent effects. Hence, the development of a model of polarized light transfer based on experimental findings was very important. Two heuristic models of polarized light transfer were proposed yielding similar results. One model describes the depolarization of a collimated beam through scattering media. This model was tested with microsphere experiments and with biological tissues. The second heuristic model is based on diffusion theory. This model was successfully tested with the Monte Carlo program. The heuristic models showed how the anisotropy parameter g is dominant in the depolarization process, large g spheres depolarizing light at a slower pace than low g spheres.

Chapter 4 describes two heuristic models. These models were tested on solutions of microspheres and on biological tissues.

What happens to light entering the superficial layer of the skin? How deep can a polarized beam penetrates before loosing all information of the incident beam polarization? Which wavelength will be better? To answer these questions several spectroscopic experiments were prepared, using three different tissue types: skin, muscle, and liver. During this experiment it was found that light depolarizes very quickly in skin, hence the images are focused in the first 300μm of the epidermis. Moreover it was found that there is a very small wavelength dependence in polarized light scattering. The use of a white light source during polarized light imaging is then appropriate. The effect of micro-domains of birefringence was also observed using a common microscope showing that the mechanisms of depolarization are linked to the number and proximity of these micro-domains.

In chapter 5, more experimental findings are developed related to polarized light transport in biological media. Mueller matrix formalism was used to obtain an approximate value of tissue retardation for skin, muscle and liver tissues.

Single cell scattering was observed with the use of an asymmetric microscope. The main goal of this experiment was the development of a tool for sizing of precancerous cell nuclei. A detailed description of this tool is shown in chapter 6.

Chapter 2
Design, testing and clinical studies of a handheld polarized light camera

2.1 Introduction

*Polarized light imaging has been used to detect the borders of skin cancer and facilitate assessment of cancer boundaries. A design for an inexpensive handheld polarized camera is presented and clinical images acquired with this prototype are shown.

The camera was built with two USB color video cameras, a polarizing beam-splitter cube and a 4x objective lens. Illumination was provided by 3 white LEDs and a sheet polarizer. Horizontal and vertical linearly polarized reflected images were processed at 7 frames per second and a resulting polarized image was displayed on screen. I compared the performances of cheap USB camera and a 16-bit electronically cooled camera. Dark noise and image repeatability were compared. In both cases the 16-bit camera outperformed the USB cameras. Despite these limitations the results obtained with this USB prototype were very satisfactory. Examples of polarized images of lesions taken prior to surgery are presented.

Polarized light imaging has been shown [3,27] to give relevant information on the borders of skin cancers that are not visible to the naked eye. Skin cancers typically originate in the superficial regions of the skin (epidermal basement membrane) where polarized light imaging is most effective. Quick assessment of skin cancer margins before Mohs surgery could guide the doctor during excision and reduce the surgery time

and patient discomfort. A number of polarized light camera systems have been used in the clinic [52,53,54] but routine use has been limited by such factors as size, weight, cost, and poor user interface. Commercially available system such as Dermlite® photo and Dermlite® platinum (3 Gen, LLC, Dana Point, CA) based on cross-polarization imaging yielded good results [55]. These systems are useful for eliminating glare and shadows from the field of view but do not provide information on the backscattered degree of polarization and superficial light scattering. More complex systems based on confocal microscopy [56,57,6] allow high resolution, and imaging the dermis to 500 μm but with much higher equipment cost and limited portability. In this chapter, a simple design is presented for a fully automatic polarized light camera system that can be held with one hand.

The camera system is composed of two USB video cameras, one polarizing beam-splitter cube and a microscope objective lens. A USB-wired mouse button on the camera allows the operator to request image acquisition. The total cost of the prototype system is 350 dollars not including the computer. The hand-held device is controlled and powered through USB connections to a laptop computer. The entire system fits in a computer bag.

2.2 Materials and Methods

The main components of the hand-held polarization system are visible in Figure 2.1.

---

* This chapter was accepted for publication Journal of Biomedical Optics, special issue on tissue optics.

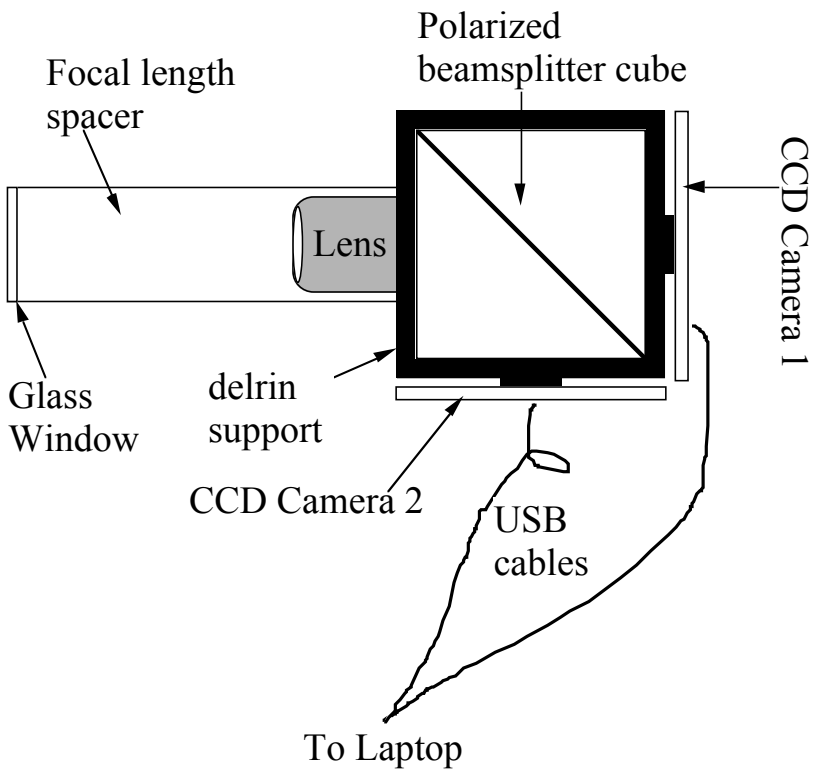

Figure 2.1. - The main components of the hand-held polarization system: polarizing beam-splitter cube, two CCD cameras and an objective lens. The CCD cameras are attached to a Delrin support enclosing the beam-splitter cube.

Two USB digital cameras (Quickcam Pro3000; Logitech Inc., Fremont, CA, USA) were disassembled and the CCD chips with support electronics were mounted on a Delrin mount that encased a broadband polarizing beam splitter cube (Melles Griot, Carlsbad, CA, USA). The beam-splitter cube had 0.01 nominal polarization extinction and was used to separate the two states of linear polarization, i.e., parallel and perpendicular to the orientation of the polarized source.

Each camera had color 1/4" CCDs capable of acquiring 30 frames/second with an image size of 640x480 pixels. An interchangeable objective lens allowed different magnifications. An objective lens (PL 4/0.1) mounted at a distance of 5 cm from the front of the beam-splitter cube was the most commonly used imaging assembly. This lens sets the camera's field of view at 1.5x1.2 cm.

The CCDs and beam-splitter cube were encased in an ergonomic plastic sphere as shown in Figure 2.2. The spherical case protected the components and supported the light source and the focal distance assembly.

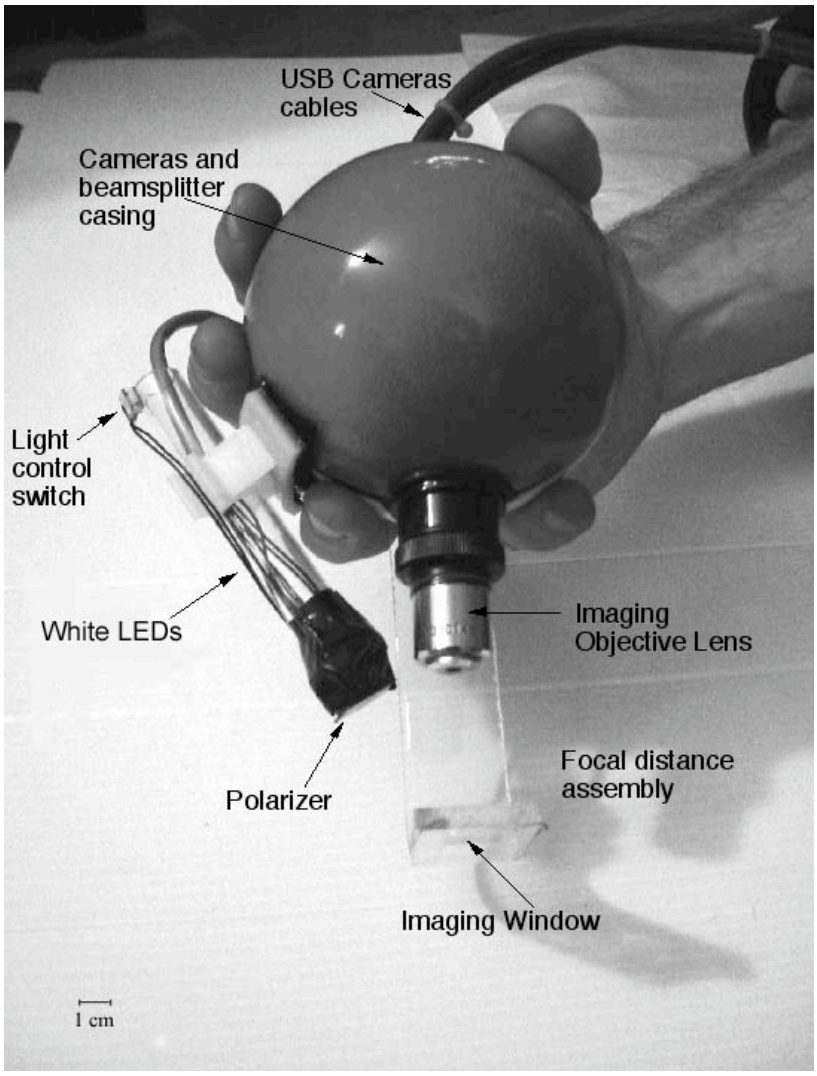

Figure 2.2. - The clinical hand-held polarization system, spherical casing protects the main components and makes the prototype easy to hold with one hand.

The polarized light source was constructed using three ultrabright white LEDs mounted on an acrylic support. The LEDs were battery powered by three AA batteries stored in a compartment in the bottom of the sphere. LED light was linearly polarized parallel to the source-sample-detector plane by a sheet polarizer of extinction 0.0001 (Hinds Inc., Portland, Oregon). The extinction of a polarizer the ratio $\frac{T_{perpendicular}}{T_{parallel}}$ when the incident state of polarization is parallel and T stands for transmission. The light source assembly was oriented at approximately 30 degree to the objective lens axis to avoid glare.

A fixed distance between the imaging plane and the cameras was maintained using an adjustable acrylic support with a mount for a glass microscope cover slip that constituted an optical flat. During operation, the glass cover slip was placed in contact with the skin. A drop of water was used to minimize tissue-air-glass reflections.

The USB cameras were connected to a 667 Mhz PowerBook laptop (Apple Computer Inc., Cupertino, CA, Model # A1001). The controlling software was written in C starting from the QuickTime Software Development Kit (SDK) [58,59]. QuickTime is a system-level code package available on most computers that support multimedia tasks. Images were acquired sequentially from two devices, Camera 1 and Camera 2. Camera parameters such as gain, shutter aperture and window orientation were modifiable through a user menu. During experiments, the gain was fixed and the shutter aperture was kept equal for both cameras. After streaming from the two cameras, every frame was decompressed and a polarized image Pol was calculated at each pixel in the red, green and blue channels:

$$Pol = \frac{parallel - perpendicular}{parallel + perpendicular} = \frac{camera1 - camera2}{camera1 + camera2} \qquad (2.1)$$

where parallel is the image from Camera 1 and perpendicular is the image from Camera 2 (Fig. 1). A composite color image was generated from the three Pol images and displayed on screen. Alternatively, either the red, green, or blue Pol image could be displayed on the screen by pressing the R, G, or B key on the computer. A second

window on the screen displayed the original color parallel image to facilitate camera positioning.

The maximum frame rate in order to display the polarized light image in a full size window (400x400) was 7 frames/sec. The frame rate could be improved by decreasing the image size. When the window was 250 x 250 pixels the frame rate increased to 15 frames/sec. The software allowed the user to average up to 20 Pol images before displaying the results on screen.

The operator could save a desired image by pressing the USB mouse button on the camera. Images were saved in the Apple PICT image format.

2.2.1. System Calibration

Experiments were conducted to test the performance of the cameras and polarizing optics.

2.2.1.1 Camera dark noise

This test was performed with the camera software shutter off. The test was performed in the dark and dark images were captured at increasing time intervals. Camera streaming was interrupted only during an image save. The camera was run for two hours although typical clinical use of the camera is much brief. The 12 captured images were processed using Matlab software and results can be seen in Figure 2.3. The mean and standard deviation was calculated for 400 x 400 x 12 = 192,000 image pixels, for the three (red, green and blue) images. The blue channel dark noise was the highest at 31 counts (standard deviation = 3 counts) where full-scale response ranged from 0 to 255 counts. The red and green channels dark noise was 15 counts (standard deviation = 3 counts).

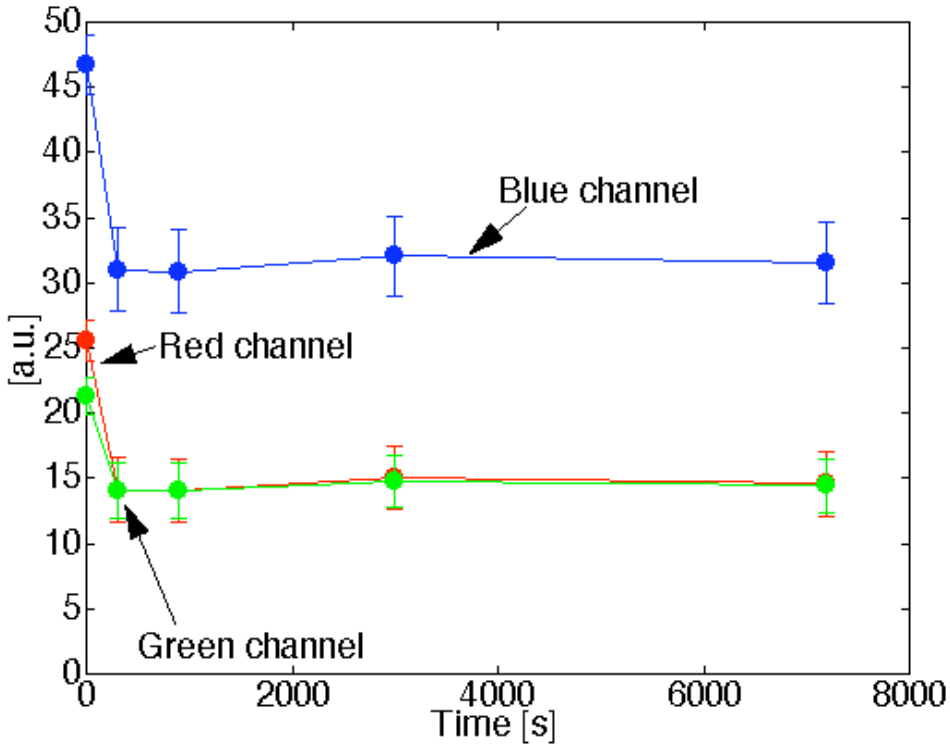

Figure 2.3. – Dark noise experiment. The average value of a dark noise image in laboratory units is plotted versus acquisition time.

For all channels, dark noise did not vary over time allowing us to subtract it from the acquired images programmatically. For comparison the same dark noise experiments were performed for a 16-bit digital camera electronically cooled camera (Roper Scientific). The dark noise was approximately 500 counts where the full-scale response ranged from 1-65453.

2.2.1.2 Camera repeatability

This test was performed to establish the pixel-to-pixel variation during multiple acquisitions. The target was a checkerboard consisting of dark black squares and bright white squares. A group of 20 images of the checkerboard was captured over a 10 minutes period. The variability from image to image was analyzed using Matlab. Different pixel locations were chosen on the images. For every pixel location in both

black and white squares the standard deviation of the 20 images was calculated and was less than 3 counts, showing the degree of image-to-image repeatability. A similar test performed with the 16-bit digital camera gave a pixel-pixel variation of 20 counts. Once again the scientific camera outperforms the cheap USB cameras. Currently small scientific 8bit digital cameras are being considered to replace the USB camera in the design of a better prototype.

2.2.1.3 Polarization optics test

The performances of the polarizing beam-splitter and source polarizer were tested in transmission mode. The polarized white light source was positioned in front of the hand-held system behind a linear polarization filter (Ealing Inc) with its polarization axis aligned parallel to the beam splitter horizontal axis. The source polarizer was rotated in 10 degree intervals. At each angle two images were collected: one with Camera 1 (Parallel) and one with Camera 2 (Perpendicular). The transmitted light beam filled most of the cameras' field of view. The pixel intensity counts of the central 100 x 100 pixels were selected to eliminate any beam shape artifact. Dark noise was subtracted from every image.

$$Pol = \frac{(camera1 - Background) - (camera2 - Background)}{(camera1 - Background) + (camera1 - Background)} =$$

$$= \frac{camera1 - camera2}{camera1 + camera1 - 2Background}$$

Results are shown in Figure 2.4.

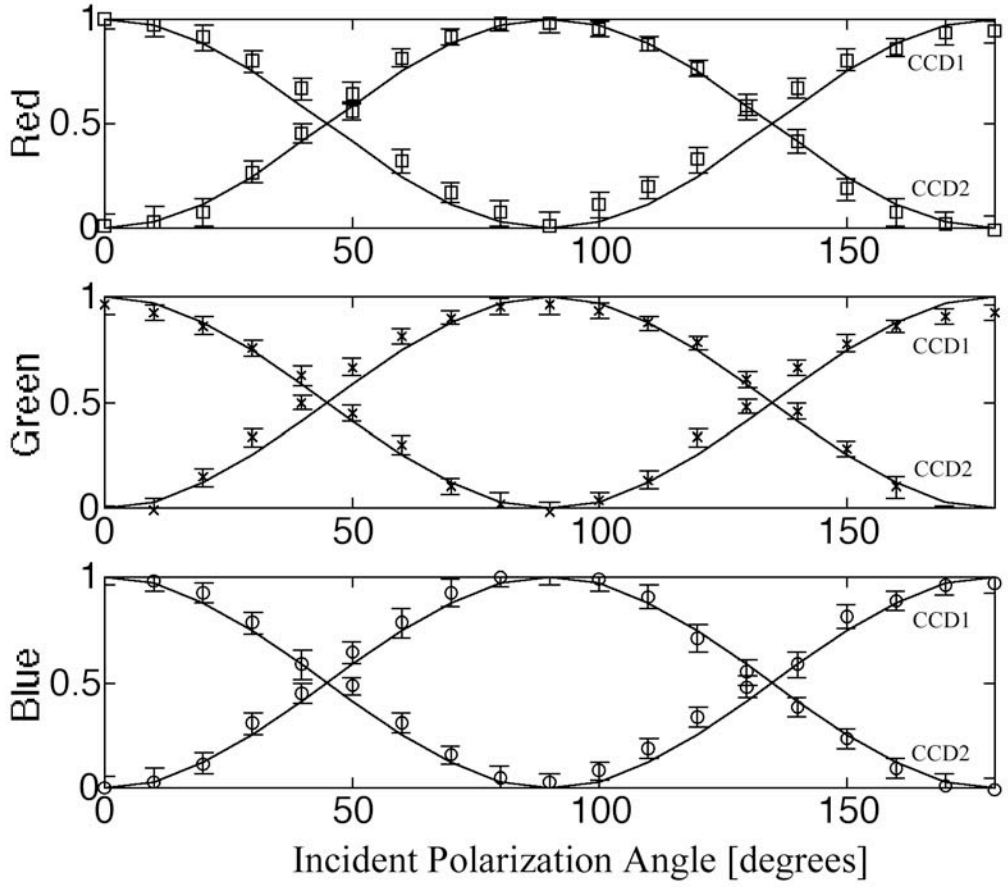

Figure 2.4. - Performance test for polarized optics. The x-axis denotes the angle of orientation of the linear polarization of the light source (0° denotes parallel to horizontal axis of beamsplitter). The y-axis denotes the detected intensity for the red, green and blue channels of the two cameras (at x-axis equal to 0°, the parallel camera 1 camera sees maximun light and the perpendicular camera 2 camera sees zero light).

The mean and standard deviation of all three camera channels, red diamonds, green crosses and blue circles are plotted versus the source polarization angle. The data was normalized by the incident 0˚ degree value for camera 1 and incident 90˚ degree value for camera 2. The experimental data was modeled with Malus' law: $I = \cos^2\theta$, where θ is the source polarization angle. The model appear as lines in Figure 2.4.

A solution of 2-μm-diameter microspheres (Fullam Inc. Latham, New York) was diluted until the scattering coefficient was equal to 29 cm^{-1}. The concentration was confirmed with a submerged optical fiber measurements of the lateral diffusion of light

from a point source as a function of distance from the source [60]. A rectangular cuvette with an optical pathlength of 1.2 cm was filled with the diluted solution. A red He-Ne laser (Melles Griot, Carlsbad, CA, USA) was used for the experiment. The laser beam had a nominal wavelength of 632.8 nm, the beam size was 2 mm and was polarized parallel to the optical bench and the horizontal axis of the beamsplitter. The beam illuminated the cuvette at 10˚ relative to the cuvette normal. The polarized system was located normal to the cuvette. The image of the polarized image of light reflected from the microsphere solution is shown on the left-hand side of Figure 2.5. The right side shows the results of a Monte Carlo simulation for this experiment. The Monte Carlo image was obtained with a code [61] that keeps track of the polarization of the photon after every scattering event.

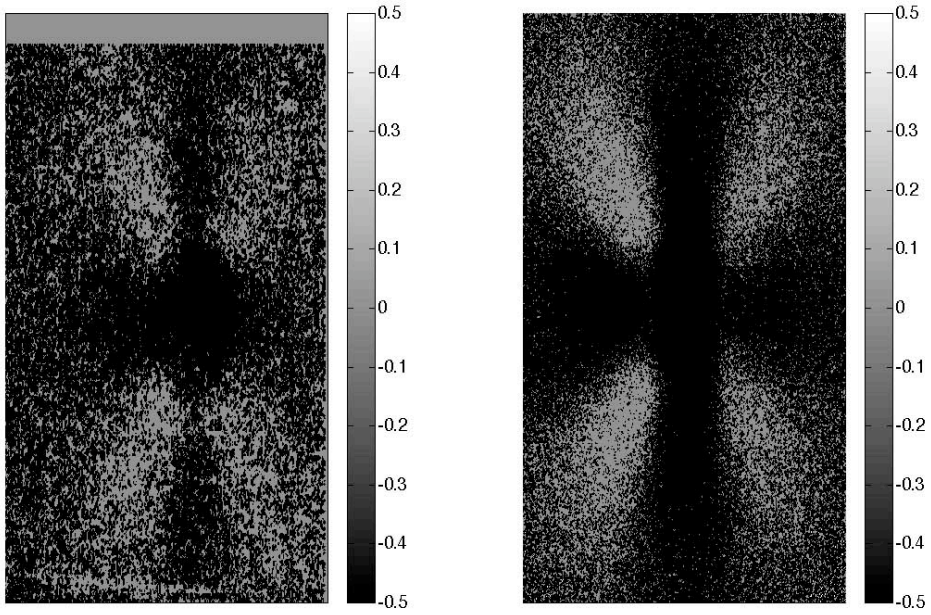

Fig. 2.5. - Monte Carlo model and experimental results of a 633-nm wavelength laser beam scattering from a solution of 2-mm microspheres in water. On the left is the image of the experimental polarized image, and on the right is the corresponding image of the Monte Carlo model.

For the Monte Carlo program, a slab-geometry was used that did not consider the air-glass interface, which might explain the discrepancies between model and experiment. The structure of the Monte Carlo program was similar to the one written by Kattawar *et al.* described in reference [62]. 10^7 photons were used for the simulation. In both images a Maltese cross scattering pattern was clearly observed and is typical of this kind of experiment.

2.3. Error analysis

To establish the reliability of the measurements an error analysis of the components of the device was performed. In Figure 2.8 a schematic of possible sources of error is shown.

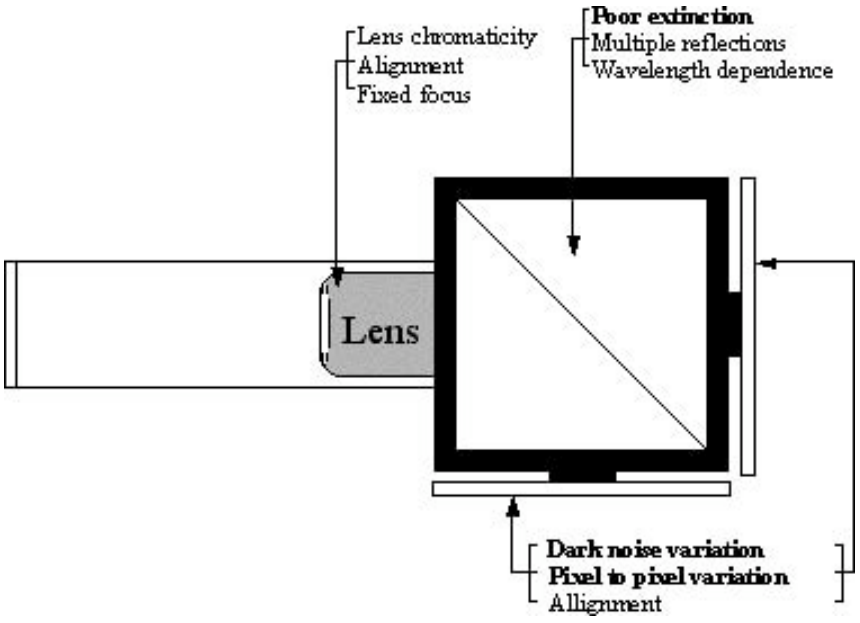

Fig. 2.6 – Error analysis of the optical components of the polarized light camera

The error related to the objective lens were identified as:

- Lens chromaticity:

- Error related to misalignment in between the camera and the lens.
- Error caused by fixed focus of the camera.

The error related to the polarizing beam splitter cube was:

- **Extinction Tp / Ts =0.01**
- Multiple reflections
- Wavelength dependence

The error related to the camera was:

- **Variability in the dark noise**
- **Pixel to pixel variability**
- Misalignment of the cameras to the beam splitter (tilt or shift)

In the previous section some of these sources of error were measured experimentally. The poor extinction of the polarizers is a major source of error in the polarized light camera and need to be further investigated. Most misalignment error were fixed in the software. Error such as chromaticity of the lens or multiple reflections in the beam splitter cube are very small compared to the other sources of error such as poor extinction, pixel to pixel variability and dark noise variation. The error analysis focuses on these three last errors: δex_{par} and δex_{per} (error due to poor extinction for the parallel and perpendicular polarizer), δdn error due to variation of dark noise, δpp error due to pixel to pixel variation.

First the uncertainty in pol (δpol) due to the poor polarizer extinction is studied.

From the definition of Pol

$$Pol = \frac{parallel - perpendicular}{parallel + perpendicular}$$

we find the uncertainty in Pol due to the poor extinction of the 2 polarizers parallel and perpendicular

$$\delta Pol = \sqrt{(\frac{\partial Pol}{\partial parallel}\delta ex_{par})^2 + (\frac{\partial Pol}{\partial perpendicular}\delta ex_{per})^2} \quad (2.2)$$

Solving these equations

$$\frac{\partial Pol}{\partial parallel} = \frac{2perpendicular}{(parallel + perpendicular)^2}$$

$$\frac{\partial Pol}{\partial perpendicular} = \frac{-2parallel}{(parallel + perpendicular)^2}$$

Finally δPol is found:

$$\delta Pol = \frac{2}{(parallel + perpendicular)^2}\sqrt{parallel^2\delta ex_{par} + perpendicular^2\delta ex_{per}} \quad (2.3)$$

The poor extinction can give an overestimate or underestimate of the parallel or perpendicular component of 10% of their total intensity. A polarized light image of a basal cell carcinoma collected with the hand-held device was used to assess the error amount.

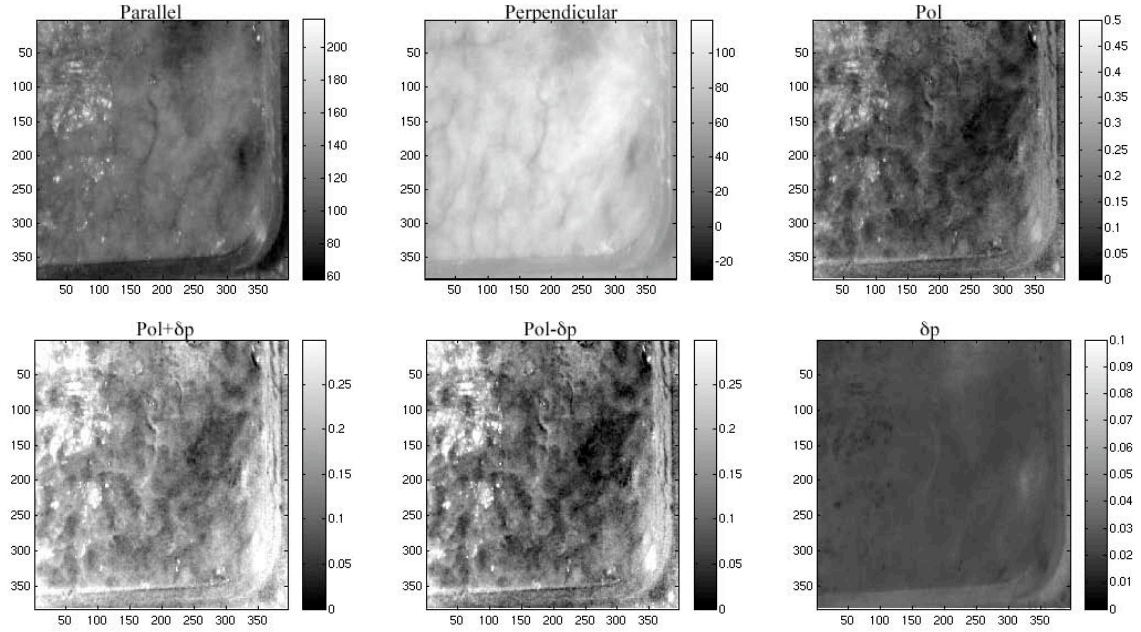

Fig 2.7 - Polarized image of a Melanoma in situ. The image in the top raw are the raw par, per and pol images. The images in the bottom raw are the images after the error calculation.

The error analysis shows that the error due to the extinction of polarizer is substantial ~ 5%. Moreover the error increases in darker areas, due to the poor signal-to-noise. Better extinction would drastically decrease the amount of error.

The error due to a pixel to pixel variation of three counts was implemented as in equation 2.3, the results for the δp are visible in Figure 2.8

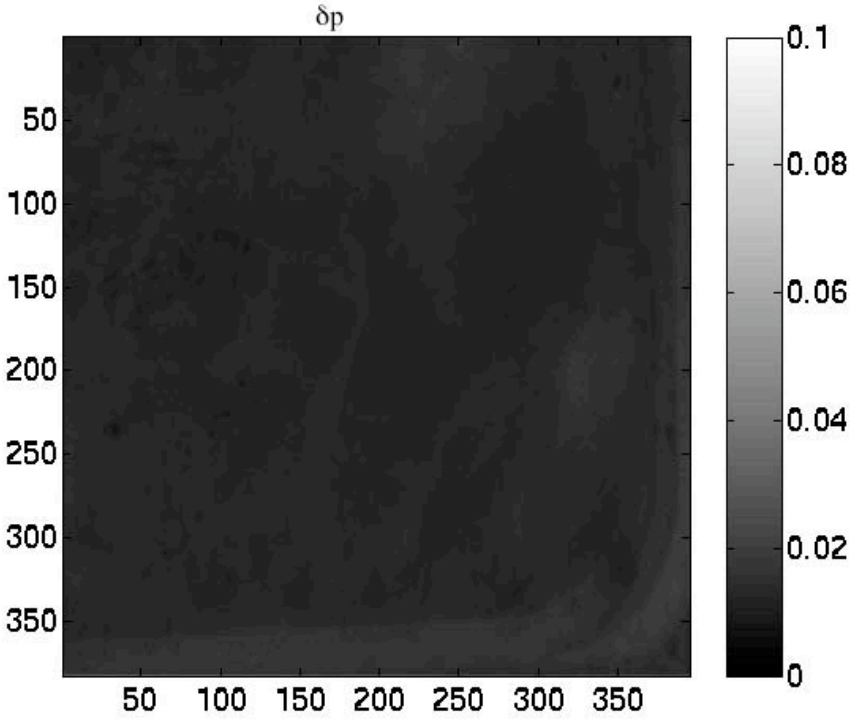

Fig 2.8 – Error due to the pixel to pixel variation.

Dark noise (DN) subtraction error (δn) was also studied. The variation in δn is equal to 2 counts.

$$Pol = \frac{parallel - Dn \pm \delta n - perpendicular - Dn \pm \delta n}{parallel - Dn \pm \delta n + perpendicular - Dn \pm \delta n}$$

this equation reduces to

$$Pol = \frac{parallel_{Dn} \pm \delta n - perpendicular_{Dn} \pm \delta n}{parallel_{Dn} \pm \delta n + perpendicular_{Dn} \pm \delta n}$$

so it was solved just as in equation 2.3

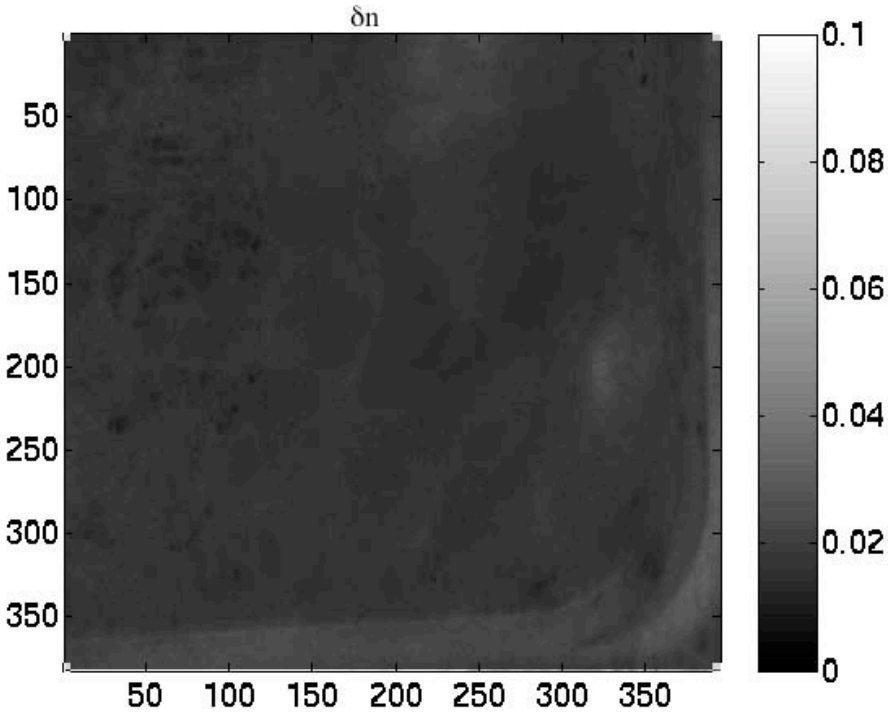

Fig 2.9 – Effect of dark noise subtraction.

Finally all the errors are added to form the total error in Pol.

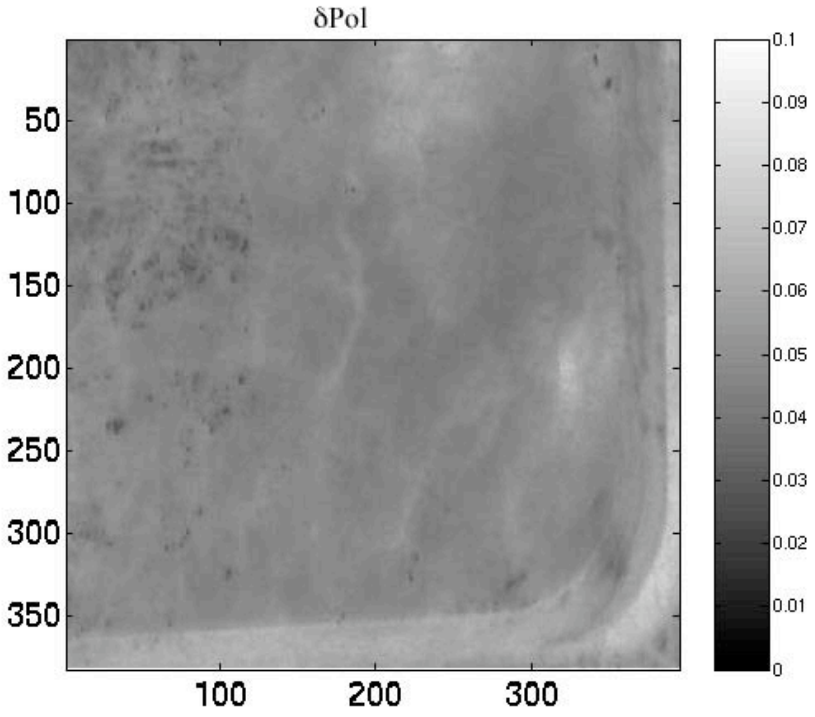

Fig 2.10 – Total error in the polarized light image.

The effect of the three studied error is significant on the polarized light image. The error is not uniform but higher in areas were the parallel and perpendicular image have both low pixel count. This effect could limit the use of the polarized camera for darker lesions. The effect of the poor extinction of the beam splitter was the larger noise, and could be easily decreased with better quality optics. The low dynamic range of the USB camera exacerbates the noise, the obvious solution to this problem is operating with higher quality cameras.

2.4. Clinical results

Clinical studies were conducted at the Department of Dermatology, Oregon Health & Science University. Consent to take part in this study was obtained from all patients. A study protocol was defined and approved by the Hospital IRB Committee. Detailed written and oral information on the protocol was given to the patients prior to

enrollment. The measurements extended the dermatoscopic procedure by an average of 10 minutes.

Images of different skin lesions were captured minutes before Mohs surgery. Figure 2.11 and 2.12 are images of a melanoma *in situ*. The image in Figure 2.11 was obtained with the parallel polarized camera and normalized by the average pixel value of an image of an epoxy standard and multiplied by 0.65 to yield reflectance. The epoxy standard had 65% calibrated reflectance across the visible wavelength spectrum corresponding to 180 counts.

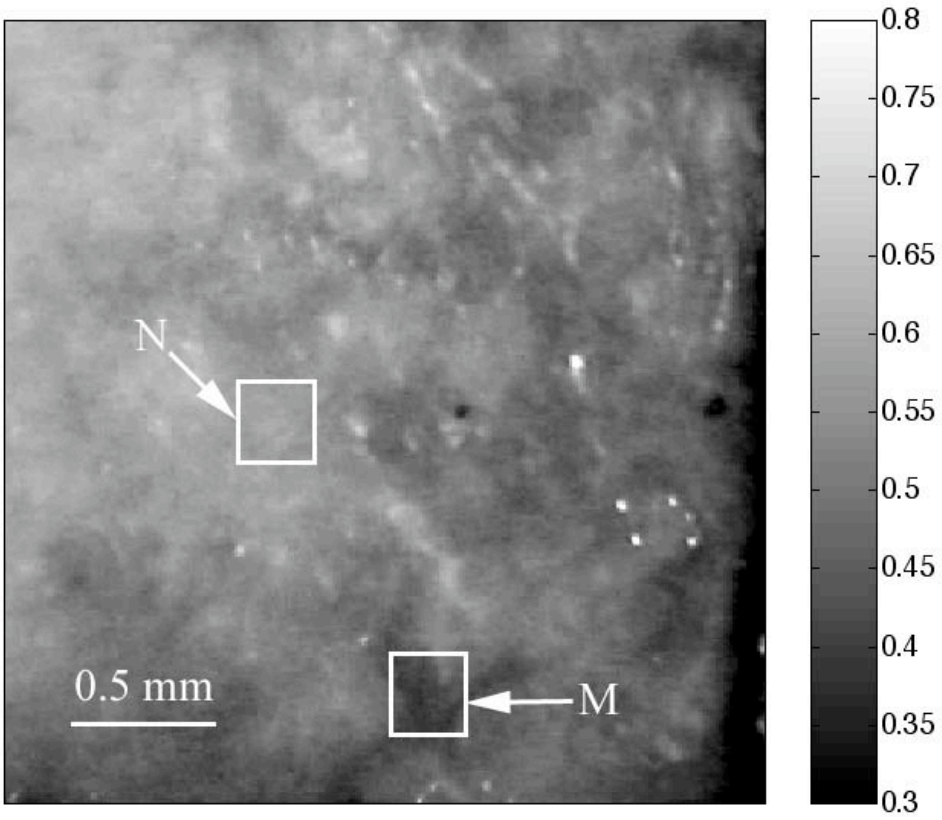

Fig 2.11 - Normal image of melanoma *in situ*. Each pixel in the image was normalized by an average value of the image of a 65% reflectance standard, then multiplied by 0.65. White squares denote two regions of interest: non-pigmented/amelanotic (N) and melanotic (M), from which data for Fig. 2.7 is collected.

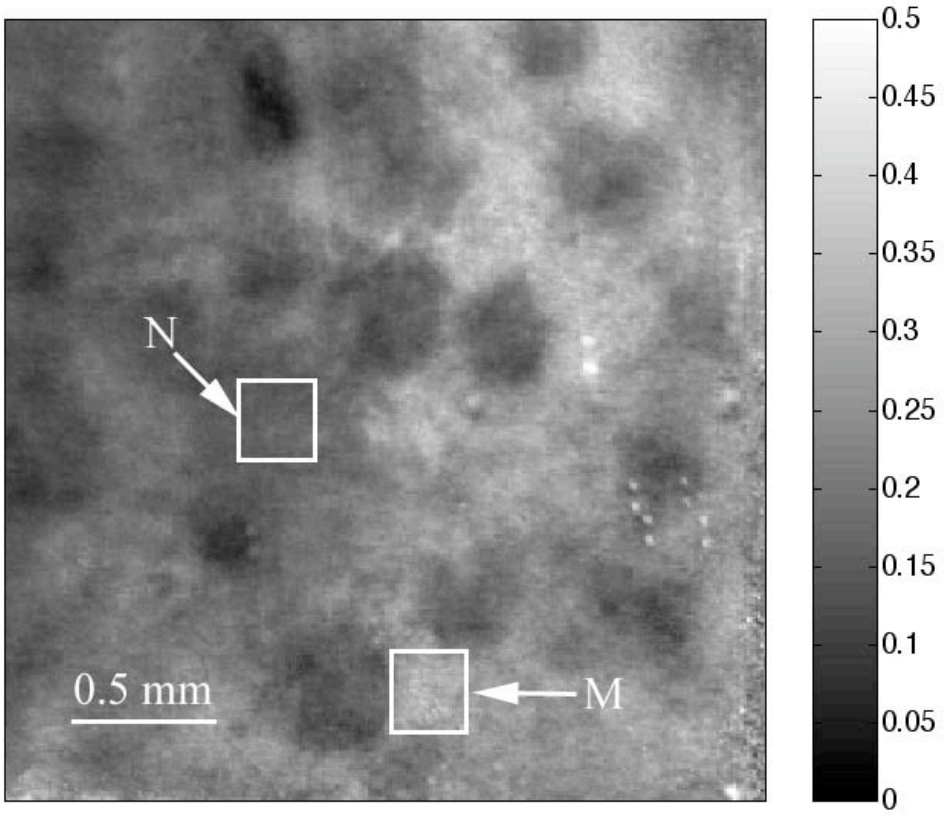

Fig. 2.12 - Polarized image of melanoma. White squares denote regions of interest, nonpigmented/amelanotic (N) and melanoma (M), from which data for Fig. 2.8 is collected.

Melanoma is comprised of abnormal proliferation of the melanocytes, the melanin pigment producing cells in the skin. Melanosomes are organelles that contain melanin. In the polarized regions with melanin pigmentation are whiter than the amelanotic regions. The whiter regions correspond to the more darkly pigmented globules in the normal image and likely correlate with cluster/nests of melanocytes. In the normal image only gray shades are visible. The increase in whiteness of the Pol pixel values is probably due to the melanosomes strongly back-scattering the incident polarized light where such backscatter retains the original status of polarization.

To illustrate the potential of the camera system two regions of interest were selected in both Figure 2.11 and 2.12. As indicated by the white squares corresponding to non-pigmented tissue (N) and melanotic (M). The pixel values for Pol and reflectance within the two regions of interest are cross-plotted in Figure 2.13. The values of reflectance for a visually normal area (N) are in the 60% range while the values for the melanotic areas (M) are around 45%. The polarized image values are higher for the Melanin regions than for the Non-pigmented regions. There is a definite separation between Non-pigmented and Melanin regions when plotted as Pol versus reflectance. R is best for discriminating the presence or absence of melanin pigment. The variability in Pol for the melanotic site provides the texture of the image and is not simply noise.

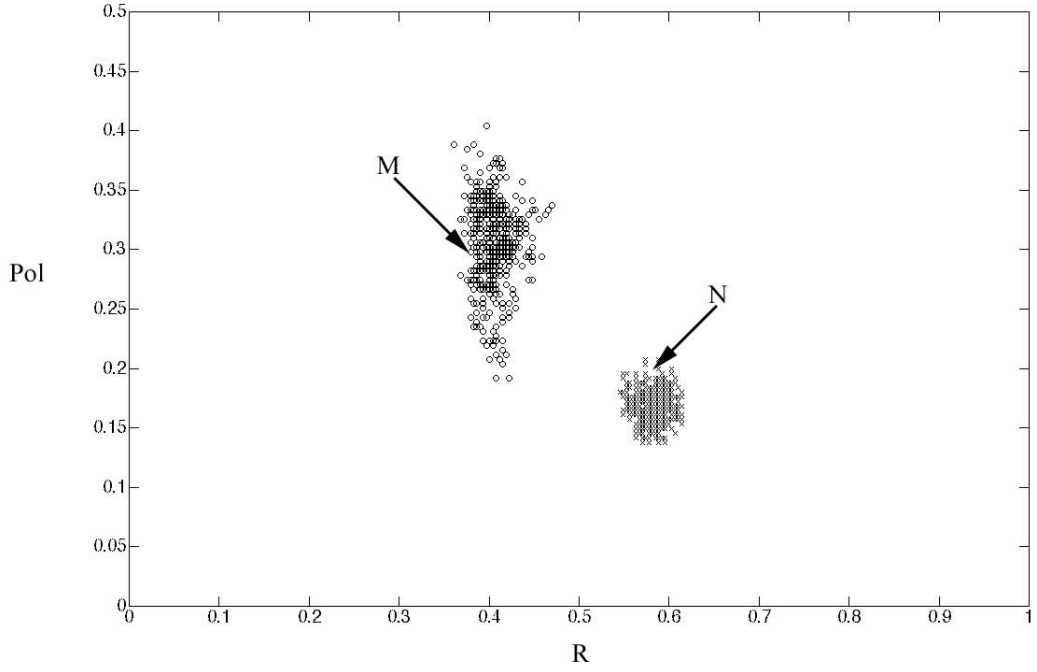

Fig. 2.13 - Plot of polarized light image values versus normal reflectance image values for two locations, one with visible melanin M and one with non-pigmented regions N. The value R was (measure of skin / measure of standard) 0.65. The standard was an epoxy block with 65% reflectance across the visible spectrum. The light source is a white LED and the reflectances of the skin and absorbtion by melanin are wavelength dependent, so the normalization by the standard gives a composite reflectance over a wavelength range.

Figure 2.14 shows images of a nevus from a forearm. The polarized image is shown on the left and the unpolarized image on the right. In the polarized image the nevus appears white and the background tissue appears dark gray. The structure of the nevus appears hilly and non-uniform. These structures are not visible in the unpolarized image.

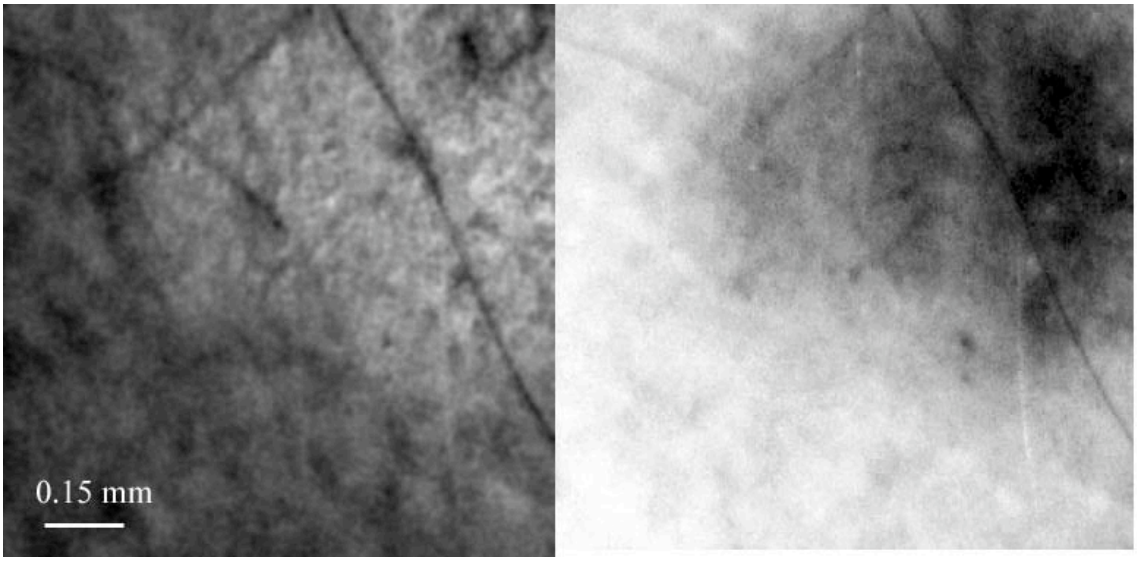

Fig. 2.14 - Image of a nevus. The apparent hilly structure of the nevus is visible in the polarized image on the left. The normal image on the right does not show this structure.

Figure 2.15 shows a squamous cell carcinoma before excision. On the left is the polarized image, and on the right is the normal image. In the normal image are visible specular (white spots) artifacts due to air bubbles between the skin and the optical flat.

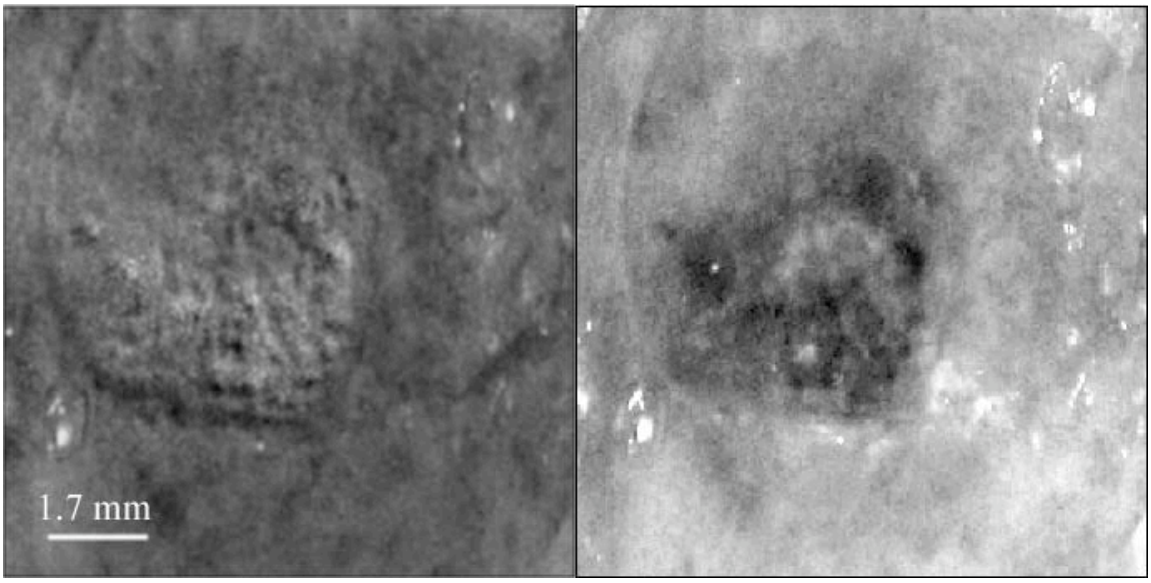

Fig. 2.15 - Polarized (left) and normal (right) image of a squamous cell carcinoma on the upper lip.

In Figure 2.15 another melanoma image is shown. In the polarized image, branches of spreading melanoma are visible, while in the normal image the melanoma is visible as darker areas.

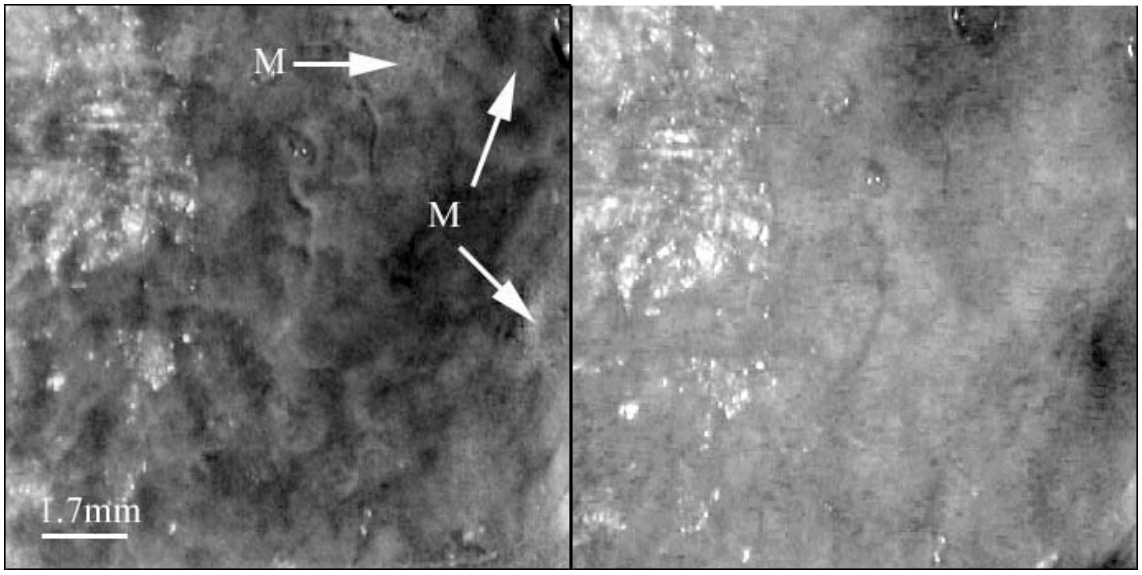

Fig. 2.16 - Image of a melanoma on a patient nose. The melanoma appears dark in the normal image (right) and white in the polarized image. The arrows show branches of the spreading melanoma. This entire area was excised during surgery.

2.4 Discussion

A hand-held device has been described that can quickly collect polarized light images. Most of the device's parts are available at any computer store. The device costs 350 dollars. Although its performance is not as good as a cooled 16-bit camera [37], the resulting images contain enough contrast for a preliminary diagnosis of skin cancer margins. Since USB video performance is continuously improving, the gap between bench-top prototypes and hand-held devices of this kind is bound to decrease rapidly.

Chapter 3
Three Monte Carlo programs of polarized light transport into scattering media

Propagation of light into scattering media is a complex problem that can be modeled using statistical tools such as Monte Carlo programs. Few Monte Carlo programs have so far included the information regarding the status of polarization of the photon before and after every scattering event. Different approaches have been followed and very few numerical values have been made available to the general public. In this chapter an overview of three different ways to build a Monte Carlo program with added polarization is given. Different groups have used the first two methods; the third method is original. A detailed description of each step necessary to reproduce the programs is offered as well as the source codes. Moreover experimental results and test cases for several sizes and densities of mono-disperse solutions of microspheres are included.

3.1 Introduction

In the last fifty years polarized light travel through scattering media has been studied by many groups and by the atmospheric optics and oceanography community in particular. An exact solution of the radiative transfer for a plane-parallel atmosphere with Rayleigh scattering was derived by Chandrasekhar in 1950 [63,64]. More complicated geometries proved too complex to be solved. Adding-doubling methods were developed

to handle the same kind of geometry for inhomogeneous atmosphere containing randomly oriented particles [65,66].

Kattawar and Plass [23] were the first to calculate the status of polarization of photons after multiple scattering using a Monte Carlo method. They later defined the degree of polarization for homogeneous layers for different solar zenith angles and optical thickness for haze and clouds [67]. Since then many papers have been published on Monte Carlo models and their applications. Kattawar and Adams used a Monte Carlo program to calculate the Stokes vector at any location in a fully inhomogeneous atmosphere-ocean system [22]. Other groups used Monte Carlo codes to analyze changes in polarization of light pulses transmitted through turbid media [68]. Monte Carlo programs were used to measure the polarization properties of dusty spiral galaxies by Bianchi *et al.* [20,69]; a detailed description of their Monte Carlo program was included. Ambirajan and Look [41] developed a backward Monte Carlo model for slab geometry with circularly polarized incident light. Martinez and Maynard [70] wrote a Monte Carlo program for the plane slab geometry and used it to study the Faraday-effect in an optically active medium [71].

Bartel and Hielsher [42] proposed a Monte Carlo model that differs from previous programs that used a local coordinate system to keep track of the polarization reference frame; the coordinate system is rotated with standard rotational matrices. They calculated the diffusely back scattered Mueller Matrix for suspension of polystyrene spheres and their results where compared with experimental results. Cameron *et al.* showed similar images calculated with a completely different Monte Carlo program [44], which satisfactorily matched the experiment. Finally Rakovic *et al.* [45] proposed a numerical method to simultaneously calculate all 16 elements of the two-dimensional Mueller matrix that compared favorably with their Monte Carlo program.

To the author knowledge none of the Monte Carlo programs mentioned above has been made freely available to the scientific community. In this paper we discuss three different Monte Carlo programs that track the status of polarization of scattered photons as they propagate in solutions of Mie and Rayleigh scatterers. The results of these

programs were compared with experimental measurements. Finally these codes are released under the *conv gnu* public license and are available for download.

The first Monte Carlo program follows the method of Chandrasekhar and Kattawar and will be called "meridian plane MC" since the photon's polarization reference plane is always relative to the meridian plane. The second Monte Carlo program uses the method described by Bartel *et al.* and will be called "Euler MC" because it uses the Euler angles used in spherical rotation problems. The third Monte Carlo program propagates a local coordinate system that is used to define the polarization state. Quaternions are used to propagate the coordinate system, hence this Monte Carlo will be termed "quaternion MC".

3.2. Standard Monte Carlo program

Monte Carlo programs are based on a technique proposed first by Metropolis and Ulam [72]. The main idea is the use of a stochastic approach to model a physical phenomenon. In the biomedical field Monte Carlo programs have been used to model laser tissue interactions, fluorescence and many other phenomena [73,74]. In Standard Monte Carlo the user is interested only in the photon weight and position; in the polarized Monte Carlo other parameters must be tracked. A flow-chart that includes both Standard and Polarized Monte Carlo main steps is shown in Figure 3.1. The gray boxes are necessary only in the Polarized Monte Carlo all other steps must be included in both approaches.

A substantial difference between the polarized and standard Monte Carlo is the choice of the azimuth and scattering angles. In the standard Monte Carlo the scattering angle θ is chosen from a phase function, the most commonly used one is the Henyey-Greenstein phase function. Generally an azimuth angle ϕ is chosen uniformly between 0 and 2π. In the polarized Monte Carlo the scattering angle and the azimuth angle are chosen by a "rejection method" (STEP C) that will be discussed in section 3.6.

An important feature of all polarization codes is tracking the reference plane used to describe the polarization of the photon. Since the status of polarization of a field is

defined with respect to the reference plane, the reference plane must be defined (STEP A) and updated for each scattering event (STEP D). Three different ways of tracking the reference plane are explored in sections 3.3, 3.4, and 3.5.

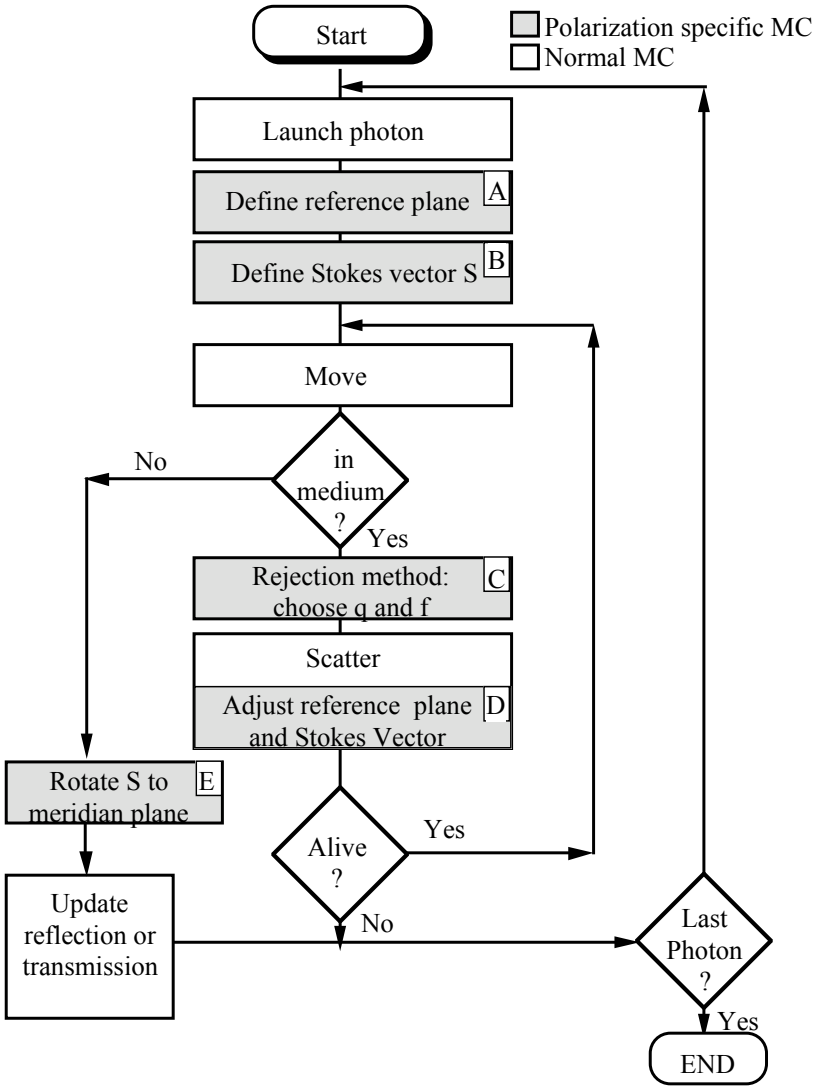

Fig. 3.1. - Flow chart of Polarized Monte Carlo program, the white cells are used in Standard Monte Carlo programs the gray cells are specific of Polarized Light Monte Carlo programs

3.3. Method 1- Meridian Planes Monte Carlo

Chandrasekhar was the first to envision the scattering of a photon from one location to the next in the form of meridian and scattering planes [63]. Figure 3.2 shows the

geometry of the problem. The photon directions, before and after scattering, are represented as points A and B respectively on the unit sphere and each photon direction is uniquely described by two angles θ and ϕ. The first angle, θ, is the angle between the initial photon direction and the Z-axis. The azimuthal angle ϕ is the angle between the meridian plane and the X-Z plane. The photon direction is also specified by a unit vector $\mathbf{I_1}$ whose elements are called the direction cosines $[u_x, u_y, u_z]$ [75]. The directions before and after scattering are called respectively $\mathbf{I}_1$ and $\mathbf{I}_2$. The unit vector $\mathbf{I}_1$ and the Z-axis determine a plane COA. This plane is the meridian plane. The incident field can be decomposed into two orthogonal components $E_{\|}$ and $E_{\perp}$, that describe the vibration of the electrical field parallel and perpendicular to the meridian plane. In this plane the polarization of the field is described with a Stokes vector **S**.

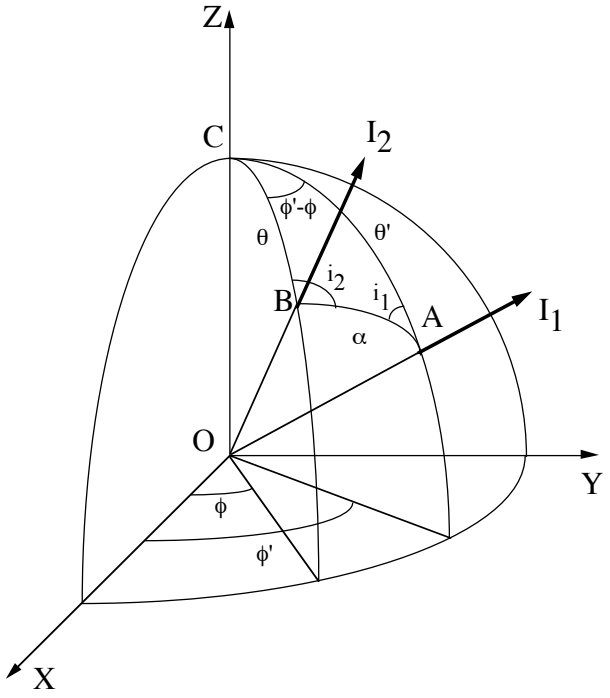

Fig. 3.2 - Meridian planes geometry. The photon's direction of propagation before and after scattering is respectively I_1 and I_2. The plane COA defines the meridian plane before and after scattering.

When a scattering event occurs a new meridian plane is created by the new direction of propagation $\mathbf{I}_2$ and the Z-axis, the Stokes vector must be transformed to calculate the new status of polarization of the field relative to the new meridian plane.

3.3.1 Launch

A photon is launched in a specific meridian/reference frame and status of polarization. For a slab geometry and perpendicular illumination the reference plane is defined by the following direction cosines $[u_x, u_y, u_z] = [0, 0, 1]$.

STEP A: The meridian plane begins with f =0, and the reference frame is initially equal to the x-z plane. The initial Stokes vector is relative to this meridian plane.

STEP B: The status of polarization of the launched photon must be defined by the user selecting an appropriate Stokes vector, **S** = [I Q U V]. The reference frame of the Stokes vector was defined in STEP A. For example if the user selects a Stokes vector **S**=[1 1 0 0] the launched photon will be linearly polarized parallel to the x-z plane.

3.3.2 Move

The photon is moved as in a standard Monte Carlo program. The photon is moved a propagation distance Δs that is calculated based on pseudo random number ζ generated in the interval (0,1], since "0" should not be included.

$$\Delta s = -\frac{\ln(\zeta)}{\mu_t} \tag{3.1}$$

where $\mu_t = \mu_a + \mu_s$, μ_a is the absorption coefficient, and μ_s is the scattering coefficient of the media. The mean free path between every scattering and absorption event is $1/\mu_t$.

The trajectory of the photon is specified by the unit vector **I** characterized by the direction cosines $[u_x, u_y, u_z]$. Following Witt [75]

$$\begin{aligned} u_x &= \mathbf{I}\cdot\mathbf{x} \\ u_y &= \mathbf{I}\cdot\mathbf{y} \\ u_z &= \mathbf{I}\cdot\mathbf{z} \end{aligned} \tag{3.2}$$

where **x** **y** and **z** are the unit vectors in the laboratory frame XYZ.

The photon position is updated to a new position [x',y',z'] with the following equations:

$$\begin{aligned} x' &= x + u_x\Delta s = I_x\Delta s \\ y' &= y + u_y\Delta s = I_y\Delta s \\ z' &= z + u_z\Delta s = I_z\Delta s \end{aligned} \tag{3.3}$$

3.3.3 Drop

As in standard Monte Carlo code the absorption of light by a dye or an absorbing material present in the scattering media is tracked by giving a weight W to a photon and updating this weight after every absorption step according to the media albedo.

$$albedo = \frac{\mu_s}{\mu_s + \mu_a}$$

3.3.4 Scatter

Three operations are necessary to scatter a photon and track its polarization; these operations are graphically described in the Figure below (3.3, 3.4, 3.5, 3.6). The **E** field is originally defined respect to a meridian plane COA. The field (visible in all Figures as blue lines) can be decomposed into its parallel and perpendicular components $E_{||}$ and $E_{\perp}$.

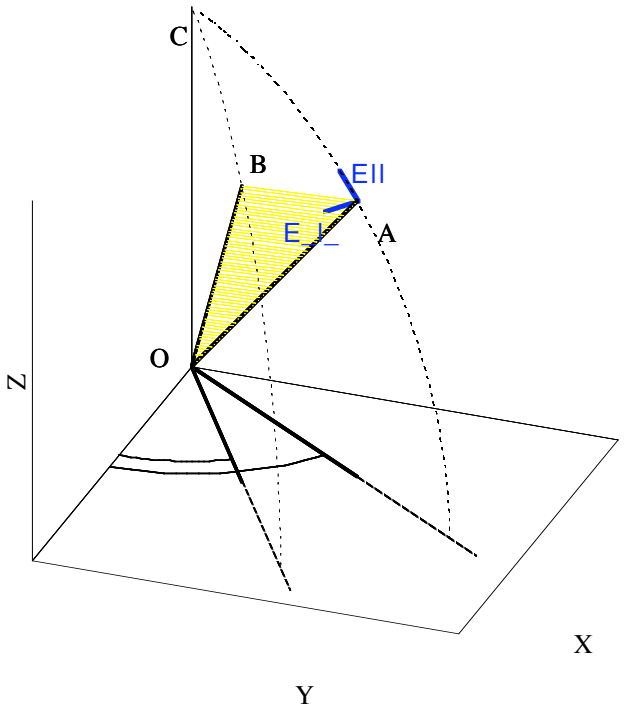

Fig. 3.3 -Visual description of the rotations necessary to transfer the reference frame from one meridian plane to the next. Initially the electrical field **E** is defined with respect to the meridian plane COA.

Once the scattering angle α and azimuth angle ϕ have been generated with the rejection method (section 6) the Stokes vector is manipulated three times. First the Stokes vector is rotated so that its reference plane is ABO, Figure 3.4; this is the scattering plane (see 3.3.4.1). This rotation is necessary because the scattering matrix, that defines the elastic interaction of a photon with a sphere, is specified with respect to the frame of reference of the scattering plane.

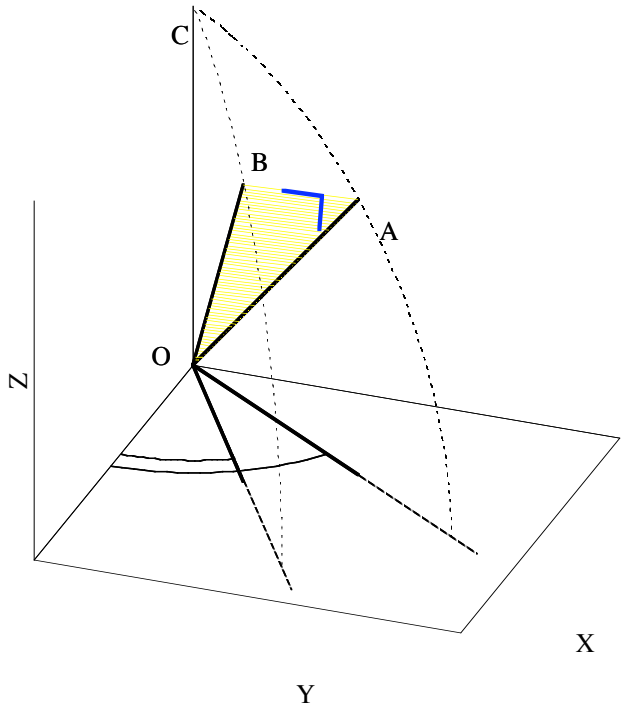

Fig. 3.4 – For a scattering event to occur the Stokes must be referenced with respect to the scattering plane BOA. The electrical field **E** (blue lines) is referenced respect to the BOA plane.

Second, the Stokes vector is multiplied by the scattering matrix that accounts for scattering of the photon at an angle α (see 3.3.4.2 below).

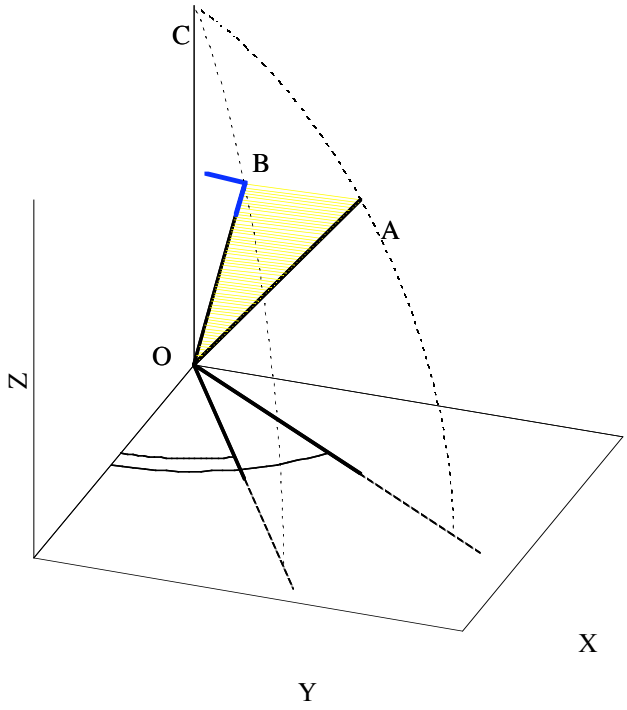

Fig. 3.5 – After a scattering event the Stokes vector is still defined respect to the plane COA. A new rotation of the reference frame is necessary to bring the reference plane back to a meridian plane COB.

Third, the Stokes vector is rotated so that it is referenced to the new meridian plane COB as shown in Figure 3.6 (see 3.3.4.3 below)..

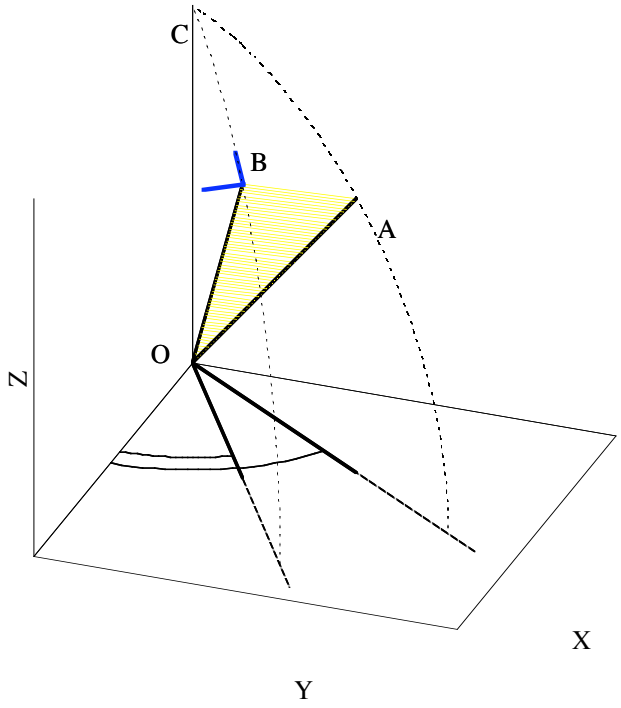

Fig. 3.6 – Second reference plane rotation. The reference frame is a meridian plane COB.

The direction cosines are updated to take into account the new direction. The new direction cosines will be called **û**. The new trajectory ($\hat{u}_x$, $\hat{u}_y$, $\hat{u}_z$**)** is calculated based on costheta, sintheta, cospsi, and sinpsi and on the current trajectory (u_x, u_y, u_z).

If $|u_z| \approx 1$:

$$\hat{u}_x = \sin(\theta)\cos(\phi)$$

$$\hat{u}_y = \sin(\theta)\sin(\phi)$$

$$\hat{u}_z = \cos(\theta)\frac{u_z}{|u_z|}$$

for all other cases:

$$\hat{u}_x = \frac{1}{\sqrt{1-u_z^2}}\sin(\theta)(u_x u_y \cos(\phi) - u_y \sin(\phi)) + u_x \cos(\theta)$$

$$\hat{u}_y = \frac{1}{\sqrt{1-u_z^2}}\sin(\theta)(u_x u_z \cos(\phi) - u_x \sin(\phi)) + u_y \cos(\theta)$$

$$\hat{u}_z = \sqrt{1-u_z^2}\sin(\theta)\cos(\phi)(u_y u_z \cos(\phi) - u_x \sin(\phi)) + u_z \cos(\theta)$$

3.3.4.1 Rotation of the reference frame into the scattering plane

The Stokes vector $\mathbf{S_1}$ defined relative to the scattering plane (ABO) is found by multiplying by a rotational matrix $\boldsymbol{R}(i_1)$ (see equation 3.8):

$$\boldsymbol{R}(i_1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2i_1) & \sin(2i_1) & 0 \\ 0 & -\sin(2i_1) & \cos(2i_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

This action corresponds to a counterclockwise rotation of an angle i_1 about the direction of propagation. The angle $i_1=\phi$ is obtained using a rejection method that will be described in section 6.

3.3.4.2 Scattering of the photon at an angle α in the scattering plane

The Stokes vector is multiplied by a scattering matrix, or Mueller matrix: $\mathbf{M}(\alpha)$. Where α is the scattering angle between the direction of the photon before scattering and after scattering. The angle α depends on the phase function, and is obtained using the rejection method of section 3.6.

$$\boldsymbol{M}(\alpha)=\begin{bmatrix} s_{11}(\alpha) & s_{12}(\alpha) & 0 & 0 \\ s_{12}(\alpha) & s_{11}(\alpha) & 0 & 0 \\ 0 & 0 & s_{33}(\alpha) & s_{34}(\alpha) \\ 0 & 0 & -s_{34}(\alpha) & s_{33}(\alpha) \end{bmatrix} \tag{3.5}$$

The parameters $s_{11}(\alpha)$, $s_{12}(\alpha)$, $s_{33}(\alpha)$, $s_{34}(\alpha)$ are calculated with Rayleigh or Mie theory. These terms are expressed as:

$$\begin{aligned} s_{11}(\alpha) &= \frac{1}{2}[|S_2(\alpha)|^2 + |S_1(\alpha)|^2] \\ s_{12}(\alpha) &= \frac{1}{2}[|S_2(\alpha)|^2 - |S_1(\alpha)|^2] \\ s_{33}(\alpha) &= \frac{1}{2}[S_2^*(\alpha)S_1(\alpha) + S_2(\alpha)S_1^*(\alpha)] \\ s_{34}(\alpha) &= \frac{i}{2}[S_2^*(\alpha)S_1(\alpha) - S_2(\alpha)S_1^*(\alpha)] \end{aligned} \tag{3.6}$$

These parameters are obtained from a Mie scattering program [76], a detailed explanation of these parameters can be found in reference [78].

3.3.4.3 Return the reference frame to a new meridian plane

The Stokes vector is multiplied by the rotational matrix $\boldsymbol{R}(-i_2)$ so that it is referenced to the meridian plane COB.

The angle i_2 was calculated by Hovenier [78] and as equal to:

$$\cos i_2 = \frac{-u_z + \hat{u}_z \cos\alpha}{\pm\sqrt{(1-\cos^2\alpha)(1-\hat{u}_z^2)}} \tag{3.7}$$

where α is the scattering angle and the plus sign is taken when $\pi < \phi < 2\pi$ and the minus sign is taken when $0 < \phi < \pi$ and $\mathbf{u}$ are the direction cosines before scattering and $\hat{\mathbf{u}}$ are the direction cosines after scattering.

In summary the Stokes vector $\mathbf{S_{new}}$ after a scattering event is obtained from the Stokes vector before the scattering event **S** using the following matrix multiplication:

$$\boldsymbol{S}_{new} = \boldsymbol{R}(-i_2)\boldsymbol{M}(\alpha)\boldsymbol{R}(i_1)\boldsymbol{S} \tag{3.8}$$

3.3.5 Photon life

The life of a photon ends when the photon passes through a boundary or when its weight W value falls below a threshold.

3.3.6 Boundaries

When the photon has reached one of the boundaries and is ready to be collected by a detector, one last rotation of the Stokes vector is necessary to put the photon polarization in the reference frame of the detector. For a photon backscattered from a slab, the last rotation to the meridian plane of the detector is of an angle ψ

$$\psi = \tan^{-1}(\frac{u_y}{u_x}) \tag{3.9}$$

The Stokes vector of the reflected photon is multiplied one final time by **R**(ψ).
For a transmitted photon the angle ψ is:

$$\psi = -\tan^{-1}(\frac{u_y}{u_x}) \tag{3.10}$$

Since Stokes vectors can be superimposed, all photons reaching a boundary can be added once they have been rotated to the reference frame of the detector.

3.4. Method 2 –Euler Monte Carlo

The second method was developed by Bartel *et al*. [42] and is based on the following idea: the photon polarization reference frame is tracked at any time via a triplet of unit vectors that are rotated by an azimuth and scattering angle according to a predefined order. In my implementation only two vectors are rotated for every scattering event **v** and **u**; the third unit vector is implicitly defined by the cross product of **v** and **u**

and is calculated only when a photon reaches a boundary. The advantages of this method is that the Stokes vector is rotated only once for each scattering event instead of twice as in the meridian plane method. Moreover the propagation of the unit vectors is straightforward and intuitive and was easier to implement. On minor drawback of this technique is the issue of the Gimbal lock that makes the rotation fail for angles exactly equal to 90˚[80].

3.4.1 Launch

The two unit vector **v** and **u** are defined in STEP A as:

$$\begin{aligned} v &= [v_x,\ v_y,\ v_z] = [0,1,0] \\ u &= [u_x,\ u_y,\ u_z] = [0,0,1] \end{aligned} \tag{3.11}$$

v and **u** define the starting Stokes vector reference plane. The unit vector **u** represents the direction of photon propagation.

3.4.2 Move

The photon is moved a distance Δs defined by equation (3.1), the new coordinates of the photon are:

$$\begin{aligned} x' &= x + u_x \Delta s \\ y' &= y + u_y \Delta s \\ z' &= z + u_z \Delta s \end{aligned} \tag{3.12}$$

3.4.3 Drop

The drop step is executed as in section 3.3.3

3.4.4 Scatter

The rejection method (section 3.6) establishes the scattering angle α and azimuth angle ϕ. The Stokes vector must be rotated by an angle ϕ into the scattering plane before

scattering can occur. This is achieved with the rotational matrix $\boldsymbol{R}(\phi)$ of the same form of equation 3.4 The actual scattering of an angle α is done multiplying the Stokes vector by the scattering matrix $\boldsymbol{M}(\alpha)$; shown in equation 3.5. After scattering the Stokes vector the reference coordinate system **v u** must be updated. The two rotations, for the angles $\phi$and α are obtained using Euler's rotational matrices [80]. There are several ways to implement these rotations; I choose the rotational matrix **ROT** in equation 3.13; this matrix accomplishes the general case of rotating any vector by an angle ψ about an axis **K**:

$$\boldsymbol{ROT}(\boldsymbol{K},\psi) = \begin{bmatrix} k_x k_x v + c & k_y k_x v - k_z s & k_z k_x v + k_y s \\ k_x k_y v + k_z s & k_y k_y v + c & k_y k_z v - k_x s \\ k_x k_z v - k_y s & k_y k_z v + k_x s & k_z k_z v + c \end{bmatrix} \tag{3.13}$$

where $\boldsymbol{K} = [k_x, k_y, k_z]$ is the rotational axis, c=*cos*(ψ), s=*sin*(ψ) and v=1-*cos*(ψ).

First the unit vector **v** is rotated about the vector **u** by an angle ϕ by multiplying **v** by the rotational matrix **ROT**(**u**,ϕ); **u** remains unchanged Figure

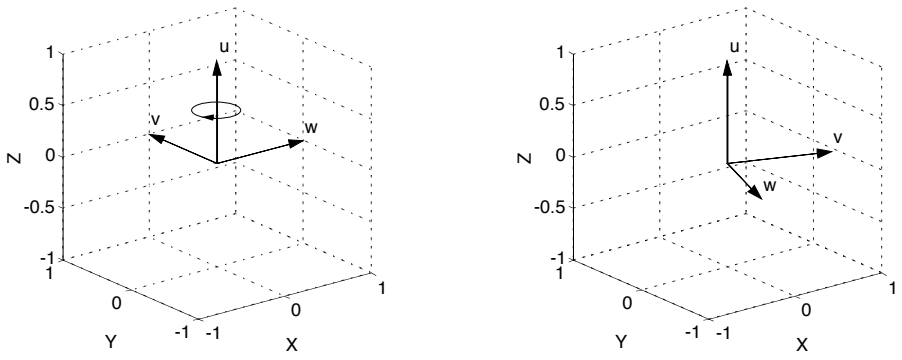

Fig. 3.7 – The triplet **w, v, u** is rotated of an angle ϕ. The rotation is about the axis **u**, left image, the rotated **w** and **v** vectors are shown on the right. **u** remains unchanged.

Then **u** is rotated about the newly generated **v** by an angle α. This is done multiplying the unit vector **u** by the rotational matrix **ROT**(**v**,α).

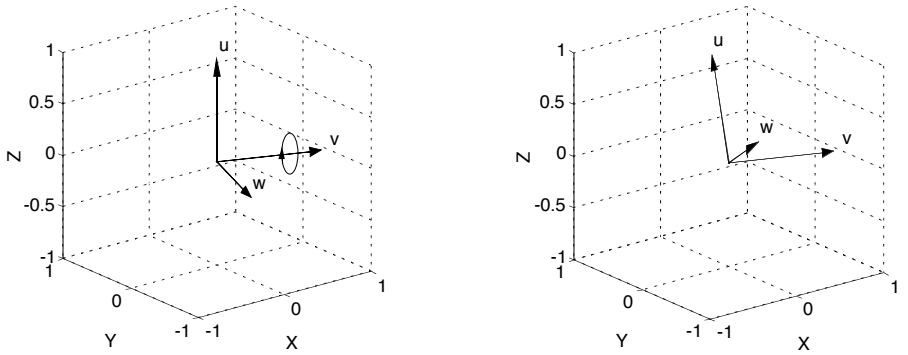

Fig. 3.8 - Second rotation on the **w, v, u** triplet. The rotation is of an angle α about the vector **v** as shown on the left. The rotated vector are shown on the right. **v** remains unchanged.

3.4.5 Photon life

A photon life ends either by absorption, when the photon weight goes below a certain threshold value or because it reaches one of the boundaries.

3.4.6 Boundaries

When the photon has reached one of the boundaries and is ready to be collected by a detector two final rotations of the Stokes vector are necessary to put the photon status of polarization in the detector reference frame.

Given a slab-geometry, for the photon exiting the media in reflection mode the first rotation is needed to return the Stokes vector to the meridian plane. To do this **w** is reconstructed as the cross product of **v** and **u**.

$$w = v \times u$$

The angle ε needed to rotate the Stokes vector into a meridian plane is given by:

$$\varepsilon = 0 \quad when \quad v_z = 0 \quad and \quad u_z = 0$$
$$\varepsilon = \tan^{-1}\left(\frac{v_z}{-w_z}\right) \quad in \quad all \quad other \quad cases \tag{3.14}$$

This rotation is about the direction of propagation of the photon, i.e. the axis **u**, Figure 3.9.

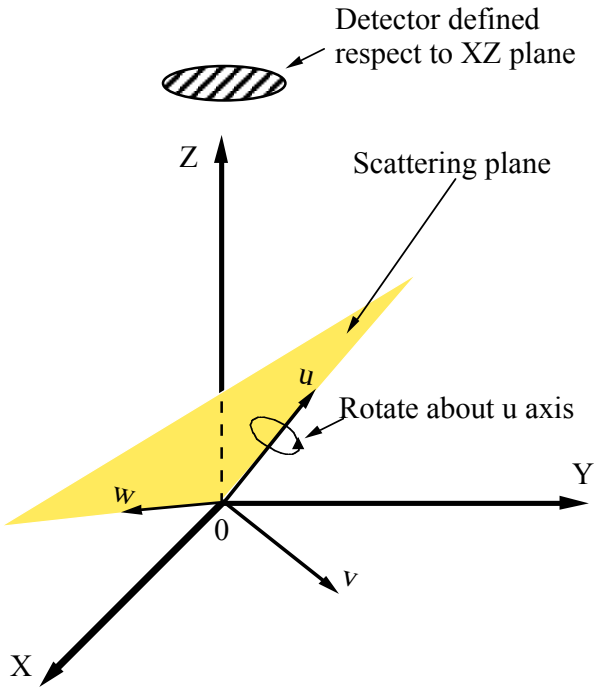

Fig. 3.9 – Rotation about an axis u of an angle ε. This rotation will bring v parallel to the Z axis and the plane w0u in a meridian plane.

After this rotation the photon reference frame is in a meridian plane, and $v_z = 0$ Figure 3.10.

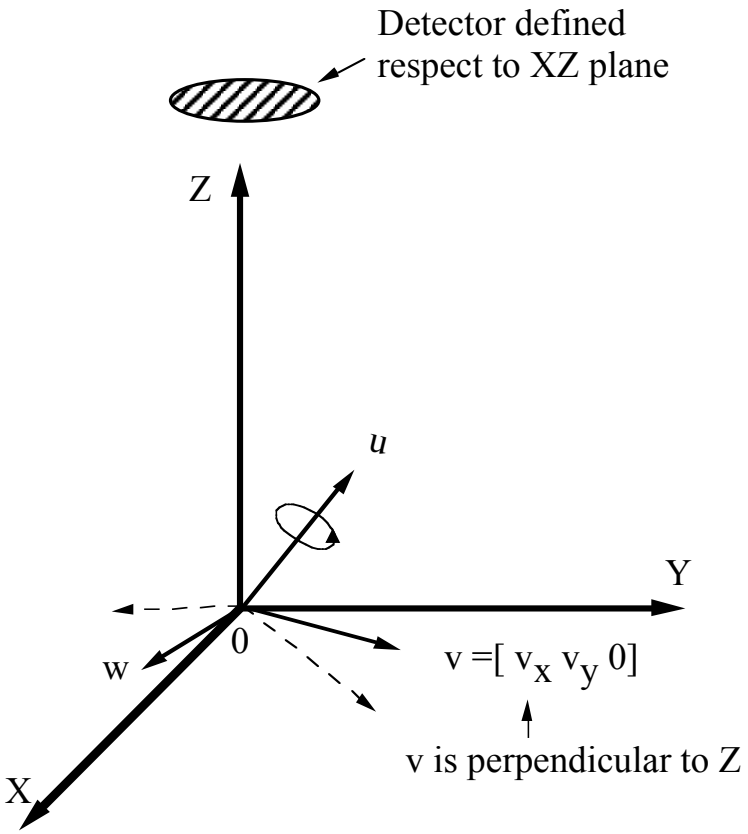

Fig. 3.10 – Effect of a rotation about an axis **u** of an angle ε on the **w** and **v** vectors. **v** is parallel to the Z axis and the plane **w0u** is in a meridian plane.

A second rotation of an angle ψ about the Z axis, Figure 3.11, will put the photon reference frame in the detector reference frame. All three reference axis wvu are affected by this rotation. ψ is calculated as:

$$\psi = \tan^{-1}\left(\frac{u_y}{u_x}\right) \tag{3.15}$$

Both **w** and **u** are in the X0Z plane, this implies that the reference plane of the photon is the same as the reference plane of the detector.

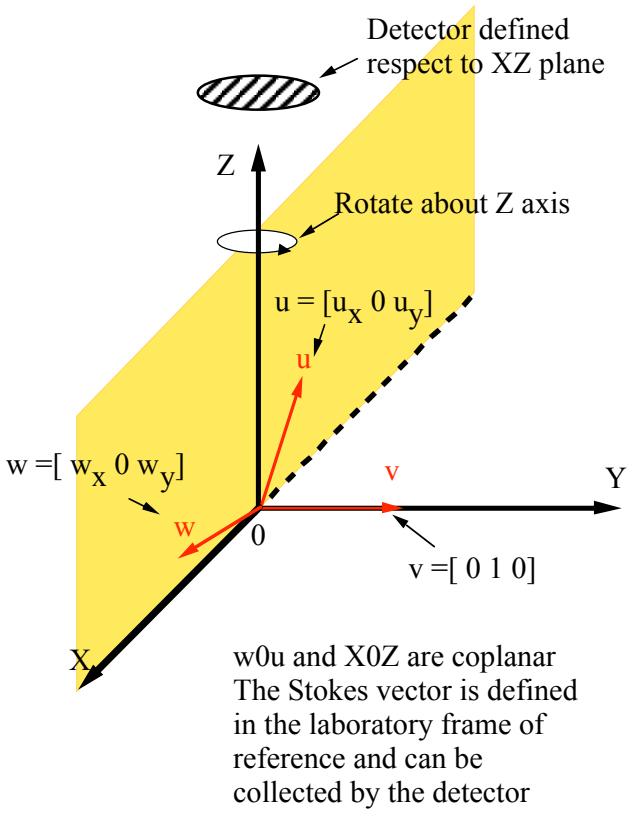

Fig. 3.11 – Rotation of the photon reference frame about the axis Z. All three vectors **u, v, w** are affected by this rotation.

In transmission mode the first rotation to the meridian plane is as in equations 3.14 the second rotation is slightly different and is equal to:

$$\psi = \tan^{-1}\left(\frac{u_y}{-u_x}\right) \tag{3.16}$$

Thus the final Stokes vector is:

$$S_{final} = R(\psi)R(\varepsilon)S \tag{3.17}$$

3.5. Method 3 –Quaternion Monte Carlo

The last method implemented is new and is based on quaternion algebra. Benoit and Clary [81] used quaternion formulation to perform calculations on molecular clusters using quantum diffusion Monte Carlo. Richartz and Hsu [82] used quaternions to

calculate the rotations and retardations of a polarized beam going through optical devices. I used quaternion algebra to track the polarization reference plane. Quaternions are not susceptible to the Gimbal lock issue. Quaternions offer an elegant and very intuitive way to handle rigid body rotations.

A quaternion is a 4-tuple of real numbers; it is an element of R^4.

$$\boldsymbol{q} = (q_o, q_1, q_2, q_3) \tag{3.18}$$

a quaternion can also be defined as the sum of a scalar part q_0 and a vector part **Q** in R^3 of the form:

$$q = q_0 + \boldsymbol{Q} = q_0 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3 \tag{3.19}$$

Quaternions can be used as rotational operators. The vector part **Q** is the rotational axis and the scalar part $q_0$is the angle of rotation. Multiplying a vector **t** by the quaternion q is equivalent to rotating the vector **t** around the vector **Q** of an angle q_0. A unit vector triplet **w, v, u** is defined as in the previous paragraph (equation 3.11). Using the property of quaternion algebra the rotation of a unit vector triplet of angles ϕ and α respectively is achieved.

The launch, move, and drop routines are the same as in the previous section, in the next paragraphs only the scattering step and the output rotations will be described.

The first rotation is about the vector **u** by an angle ϕ. First generate the quaternion q_ϕ

$$q_\phi = \phi + \boldsymbol{u} = \phi + \boldsymbol{i}u_x + \boldsymbol{j}u_y + \boldsymbol{k}u_z \tag{3.20}$$

and then multiply the quaternion by the unit vector **v**; **u** remains unchanged. The second rotation is by an angle α about the axis **v**. This is done generating the quaternion q_θ

$$q_a = \alpha + \boldsymbol{v} = \alpha + \boldsymbol{i}v_x + \boldsymbol{j}v_y + \boldsymbol{k}v_z \tag{3.21}$$

and multiplying it by the unit vector **u**.

These steps are repeated for every scattering event. At the boundaries the last aligning rotations are the same as in Euler Monte Carlo.

3.6. Rejection Method

A fundamental problem in every Monte Carlo program with polarization information is the choice of the angles α (angle of scattering) and ψ (angle of rotation into the scattering plane). These angles are selected based on the phase function of the considered scatterers and a rejection method. A detailed description of how to implement a rejection method can be found in [45], and I will summarize the main steps here. The phase function $P(\alpha,\psi)$ of spheres for incident light with a Stokes vector $S_o = (I_o, Q_o, U_o, V_o)$ is

$$P(\alpha,\psi) = s_{11}(\alpha)I_o + s_{12}(\alpha)[Q_o\cos(2\psi) + U_o\sin(2\psi)] \tag{3.22}$$

This phase function has a bivariate dependence from the angles α and ψ.
For unpolarized light where $S_o=(1,0,0,0)$, the phase function becomes a function of just the single variable α.

$$P(\alpha) = s_{11}(\alpha) \tag{3.23}$$

Hopcraft *et al.* [83] graphically showed how the phase function (3.19) depends on both α and ψ for linearly polarized light. They pointed out that the phase function is not axis-symmetric for linearly polarized light. This is visible in equation 3.22. If the Q or U components of the Stokes vector are not both zero the phase function depends both on α

and ϕ. For circularly polarized light, the phase function is axis-symmetric since there is no ϕ dependence.

My simulations of phase function intensities for linear and circular incidence agree with the simulations done by Hopcraft *et al.* and can be seen in Figure 3.11. These simulations were done with Mathematica® software package and show the Mie phase function for linear polarized incidence and circular polarized incidence. The asymmetry of the phase function along the z-axis for linear incidence is evident. In this Figure the phase function shows a strong dip at 90˚. A small sphere size (diameter = 0.01μm) was used in both graphs with index of refraction 1.59.

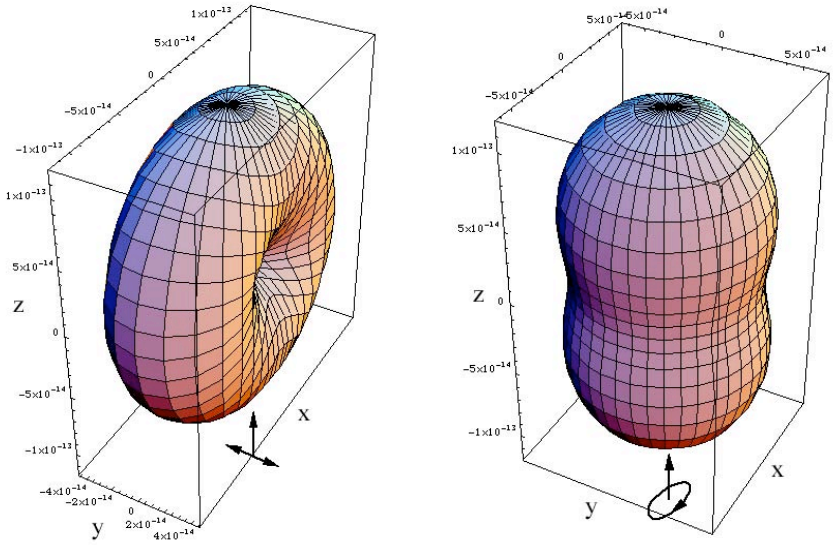

Fig. 3.11- Left Figure shows the Mie phase function for incident linear polarized light (Stokes vector [1,1,0,0]) and the right Figure is the phase function for incident circular polarized light (Stokes vector [1,0,0,1]). The sphere diameter is 0.01μm, the incident wavelength is 543 nm, the index of refraction of the sphere is 1.59.

The rejection method [84] may be used to generate random variables with a particular distribution. For functions of a single variable, two random numbers are generate. First, P_{rand} is a uniform random deviate between 0 and 1. Second α_{rand} is generated uniformly between 0 and π. α_{rand} in (3.23) to obtain $P(\alpha_{rand})$.

The angle α_{rand} is accepted as the new scattering angle if $P_{rand} \leq P(\alpha_{rand})$. If $P_{rand} > P(\alpha_{rand})$ α_{rand} and P_{rand} are generated and the test is repeated.

When an angle α_{rand} is accepted a similar process is implemented to calculate a new angle ψ_{rand}. For bivariate distribution like equation 3.22, three random numbers are generated α_{rand} P_{rand} and ψ_{rand}. The angle ψ is uniformly distributed between 0 and 2π. If $P_{rand} \leq P(\alpha_{rand}, \psi_{rand})$ then both α_{rand}, ψ_{rand} are accepted as the new angles.

3.7. Results

The results of the Monte Carlo program just described were tested against other available simulations, such as Adding Doubling code and experimentally both in transmission and reflectance mode. Mono-disperse solutions of latex microspheres in slab geometry were considered in most cases.

3.7.1 Comparison with Evans Code

In 1991 Evans designed an adding doubling code [65] that can calculate both the radiance and flux of a polarized light beam exiting the atmosphere. The atmosphere can be modeled as a disperse solution of microspheres. One important feature of Evans' code is the way in which the flux is calculated: when a photon exits the atmosphere either in reflectance or transmission mode the values of the corresponding Stokes vector are summed without relating the polarization to a single detector reference plane. This is equivalent of having several detectors oriented to the meridian planes in different locations on the unit hemisphere. Most users use a single fixed detector to collect all the exiting photons, nevertheless I can model Evans program geometry with a simple

modification to the meridian plane code by eliminating step E (Figure 3.1) from the program. This prevents the last rotation to the detector meridian plane.

Simulations were conducted considering a plane parallel slab of thickness L = 4/mus, with absorption coefficient μ_a=0, and an unpolarized incident beam (Stokes vector [1 0 0 0]) of wavelength λ = 0.6328 nm. A typical script run with the Evans' RT model is shown below.

```
C  WAVELENGTH=  0.6328000E+00
C  COATED_SPHERE=N
C  REFRACT_INDEX=  0.0000000E+00  0.1590000E+01
C  RADIUS_RANGE=   0.5000000E-01   0.5000000E-01
C  NUMBER_INTEGRATION_STEPS=   0
C  GAMMA_DIST_A=  0.1000000E+01
C  GAMMA_DIST_B=  0.0000000E+00
C  GAMMA_DIST_ALPHA=  0.0000000E+00
C  GAMMA_DIST_GAMMA=  0.0000000E+00
0.1714364E+00    EXTINCTION
0.1714364E+00    SCATTERING
1.0000000    SINGLE SCATTERING ALBEDO
10    DEGREE OF LEGENDRE SERIES
0    1.00000000  -0.49994576   0.00143192  -0.00847755   1.00000000   0.00143192
1   -0.00170337   0.00256833   1.49996185  -0.00009746  -0.00170337   1.49996185
2    0.49995219   0.49993394  -0.00570367   0.00847701   0.49995219  -0.00570367
3   -0.00256859  -0.00256869   0.00000215   0.00009746  -0.00256859   0.00000215
4    0.00001181   0.00001181   0.00000016   0.00000053   0.00001181   0.00000016
5    0.00000036   0.00000036   0.00000000   0.00000000   0.00000036   0.00000000
6    0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
7    0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
8    0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
9    0.00000000   0.00000000   0.00000000   0.00000000   0.00000000   0.00000000
10    0.00000000   0.00000000   0.00000000  0.00000000   0.00000000   0.00000000
```

This script corresponds to a 0.6328 nm waveleght of 0.5μm radius and 1.59 indec of refraction. The slab size was 4/mus.

In the Monte Carlo program the number of photon launched was 10^6.

The value of reflectance and transmittance through the slabs are shown in the following tables:

Diameter (nm)	Evans I	This code I	Evans Q	This code Q
10	0.6883	0.6886	-0.1041	-0.1042
100	0.6769	0.6769	-0.1015	-0.1009
1000	0.4479	0.4484	0.0499	0.0499
2000	0.2930	0.2931	0.0089	0.0088

Table 3.1 - Reflectance mode, comparison between Evans adding-doubling code and the meridian plane Monte Carlo program. The results do not include the final rotation for a single detector.

Diameter (nm)	Evans I	This code I	Evans Q	This code Q
10	0.33126	0.31133	-0.01228	-0.01233
100	0.32301	0.32305	-0.12844	-0.12770
1000	0.55201	0.55152	0.02340	0.02348
2000	0.70698	0.70689	0.01197	0.01205

Table 3.2 - Transmission mode, comparison between Evans adding doubling code and the meridian plane Monte Carlo program. The results are not corrected for a single detector.

3.7.2 Experimental validation

The first microsphere experiment is shown in Figure 3.6. A small glass container (7x7x4 cm) and wall thickness equal to 0.25 cm, was filled with a solution of 0.5 μm microspheres (Duke Scientific Corporation, Palo Alto, CA). The microsphere solution was diluted until its scattering coefficient was equal to 10.53 cm^{-1}. The index of refraction of the microspheres were 1.59. A Helium Neon Laser (Melles-Griot, Carlsbad, CA), at 632.8 nm and 1.2 mm beam-size was oriented so that the beam was parallel to the

face of the tank. The beam was incident normal to the side of the tank and 1cm from the front of the tank. The beam was polarized parallel to the optical table with a polarizer (Ealing, Rocklin, CA). A 12-bit digital camera (MicroMax, Roper Scientific, Trenton, NJ) was positioned perpendicular to the sidewall of the tank and focused on it. An analyzer was located between the camera and the tank. The analyzer was sequentially rotated parallel and perpendicular to the incident beam.

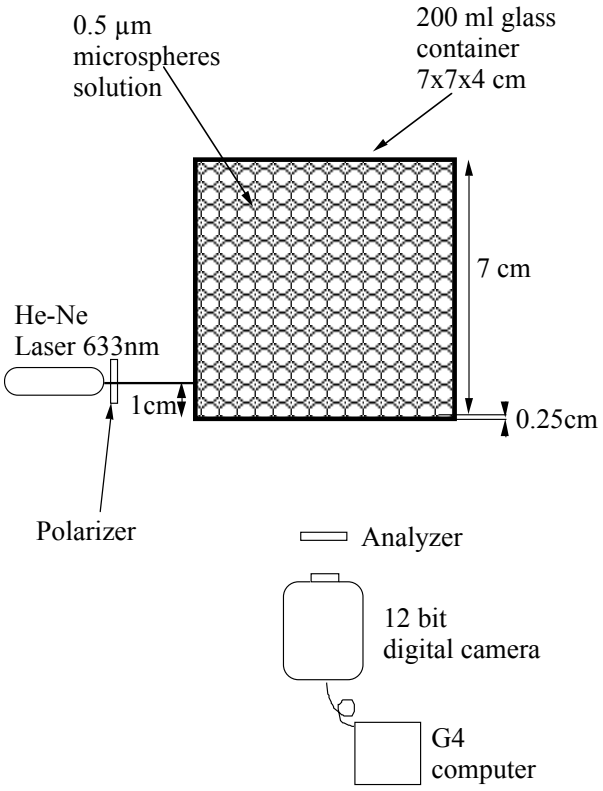

Fig. 3.12 -Experimental setup, top view. Polarized images of light scattered by a microspheres solution were acquired with a digital camera. Laser light (632.8nm) was polarized and the scattered light was passed through an analyzer whose orientation was either parallel or perpendicular to the incident beam.

Monte Carlo simulations were conducted for the same microsphere density and size. The geometry of the experiment was implemented in the code without Fresnel reflectance, so that the photon was killed when it reached the walls of the tank. The number of photons launched was one million, the simulations were conducted on a Pentium 4, 1.7 Ghz running Linux Red Hat 7.1. Results of both experimental and

calculated images were normalized to the image maximum value, a background noise of 250 counts was subtracted from the experimental images.

Figure 3.13 shows the results for the experiment conducted with polarizer and analyzer parallel to the optical table and the corresponding Monte Carlo result is shown.

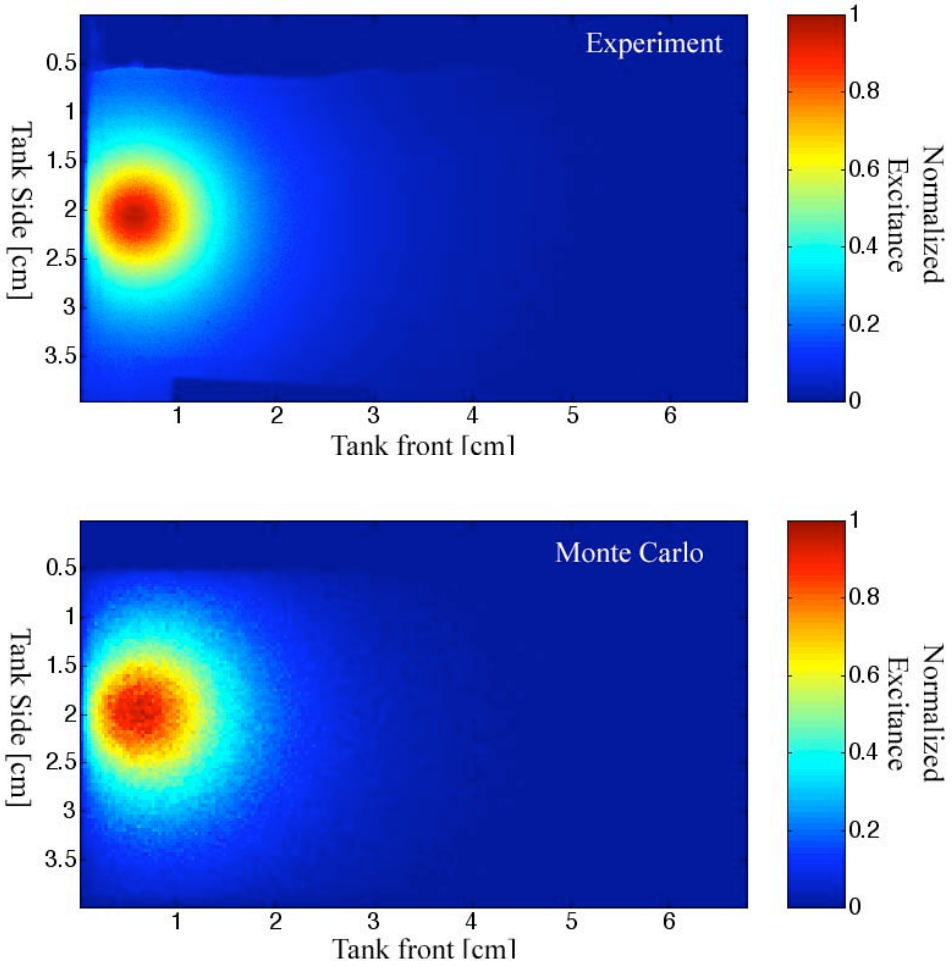

Fig. 3.13-Top image is the experimental results obtained with polarizer and analyzer oriented parallel to each other, the bottom image is the Monte Carlo simulation.

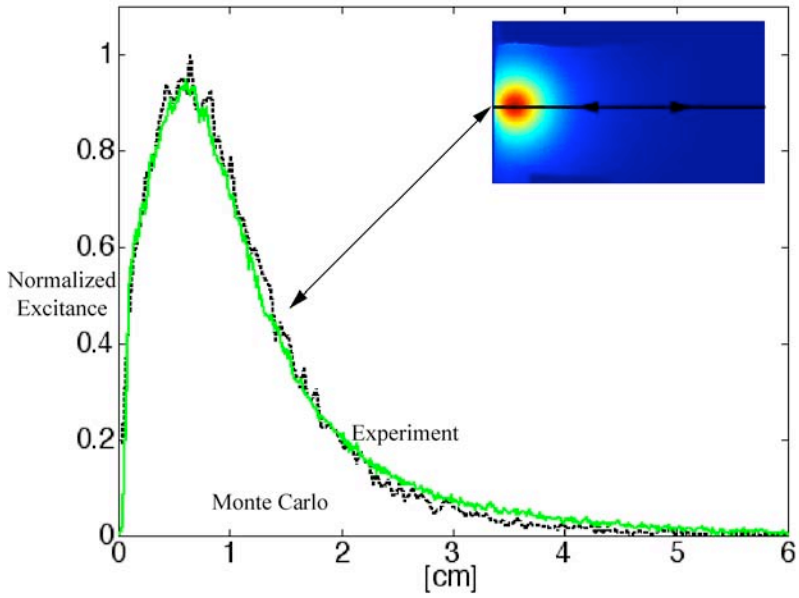

Fig. 3.14 - Comparison of excitance for experiment (green) and Monte Carlo results (black dashed) for parallel image. This intensity profile corresponds to the centerline of the axis of irradiation.

To quantitatively compare the numerical and the experimental results the normalized excitance values in Figure 3.13 on the axis of irradiation (2cm) are shown in Figure 3.14.

Figure 3.9 shows the experimental and calculated results obtained with the polarizer oriented parallel to the optical table and the analyzer perpendicular to it. The pattern is more extended longitudinally. A more diluted sample would show two distinct lobes.

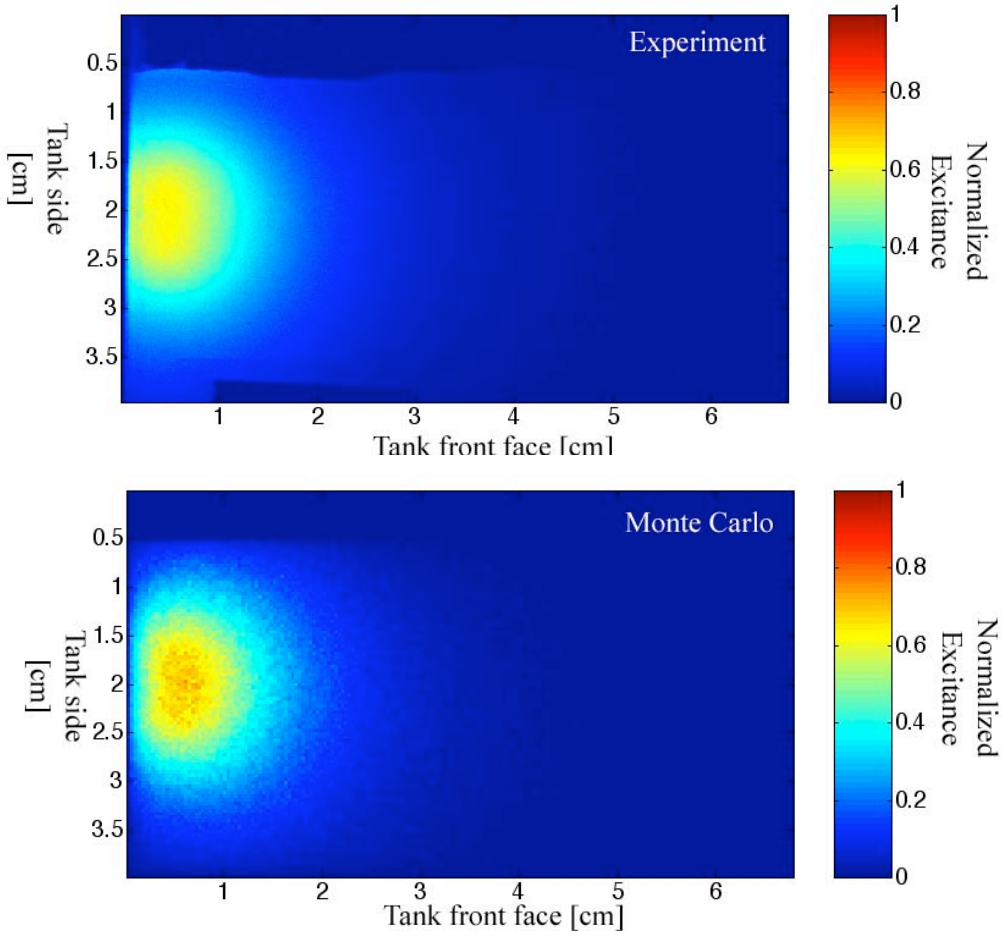

Fig. 3.15 - Top image is the experimental results obtained with the polarizer oriented parallel to the optical table and analyzer oriented perpendicular to the optical table, the bottom image is the corresponding Monte Carlo simulation. Both images were normalized by the maximum value of the parallel image.

3.7.3 Mueller Matrix of backscattered light from microspheres solutions

Monte Carlo simulations with polarization are often presented in Mueller Matrix form. Both Bartel *et al.* [42] and Cameron *et al.* [44] showed results of runs conducted with mono-disperse solutions of microspheres of diameter 2.02 μm, index of refraction 1.59 and index of refraction of water 1.33. I was able to model both their experiments and reproduce their results with all three Monte Carlo programs, Figure 3.16; during this process I realized that many of the images are symmetrical and if symmetry is broken problems could arise.

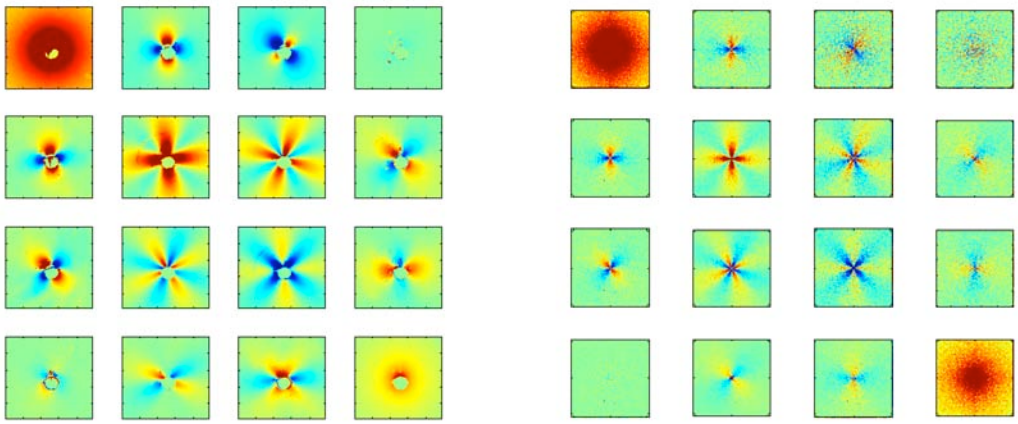

Fig. 3.16 – Comparison of Mueller matrix results for light backscattered from solutions of microspheres. The image on the left is an experimental results of Cameron *et. al.* the image on the Mueller matrix obtained with these Monte Carlo programs.

A particularly challanging illumination pattern is shown in Figure 3.18. This asymmetric experiment exposes modelling problems that are hidden in the perpendicular illumination experiment. Furthermore this problem was of particular interest since in the clinical work [37] the samples were illuminated with polarized light at an angle θ to avoid glare.

The experimental set up is shown in Figure 3.17. A mono-disperse solution of microspheres was illuminated at an angle θ~30˚ by a laser beam of wavelength 632.8 nm, and beam-size 1.2 mm. The incident beam was polarized with a linear polarizer (Ealing, Rocklin, CA) whose orientation was parallel to the plane created by laser sample and camera (H horizontal) perpendicular to that plane (V vertical), and at +45 degrees to that plane(P). The P orientation was achieved rotating the linear polarizer counterclockwise 45 degrees looking into the incoming beam. The sample was a solution of 2.0μm diameter microspheres with index of refraction 1.59 and media index of refraction 1.33 (Duke Scientific Corporation, Palo Alto, CA). Once diluted the microsphere solution had a scattering coefficient equal to 11 cm^{-1}.

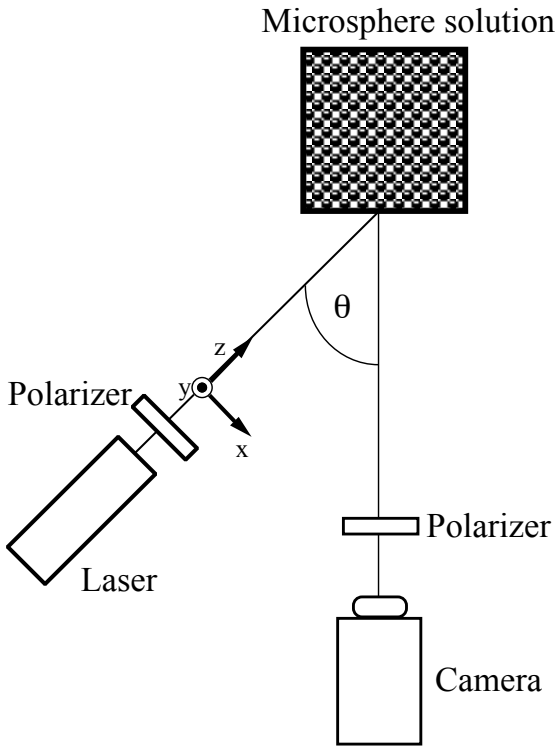

Fig. 3.17 - Experimental apparatus. I studied backscattering from a solution of microspheres (diameter 2μm) when the polarized illumination is at an angle q. Three different light source polarizer orientations (H V P) were used as well as three detectors polarizer orientation (H V P). Images were combined to produce 9 of the 16 element of a Mueller matrix.

A set of 9 images was collected with a 12-bit digital camera (Roper Scientific, Trenton, NJ). For every source polarization, the detector was oriented to H, V, and P. The P orientation was generated by rotating the linear polarizer or analyzer counterclockwise 45 degrees when looking at the incoming beam. The 9 collected pictures $H_{source}H_{detector}$, $H_{source}V_{detector}$, $H_{source}P_{detector}$, $V_{source}H_{detector}$, $V_{source}V_{detector}$, $V_{source}P_{detector}$, $P_{source}H_{detector}$, $P_{source}V_{detector}$, $P_{source}P_{detector}$, were combined as in Yao and Wang [85] to yield the first 9 terms of the backscattered Mueller matrix Figure 3.12.

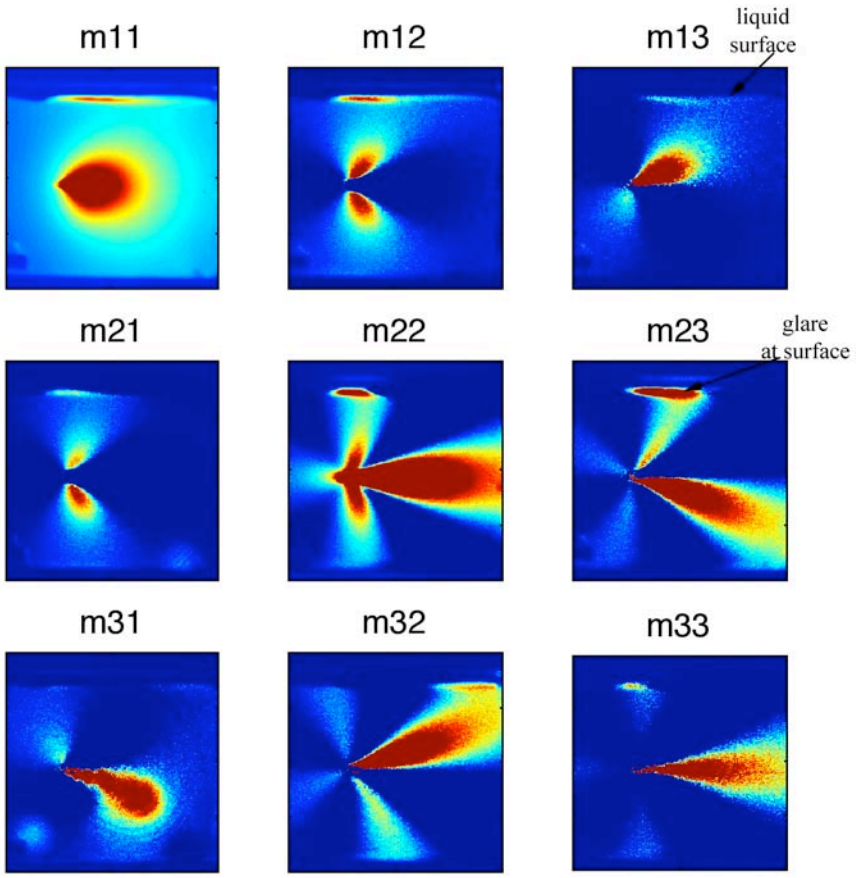

Fig. 3.18 - First 9 of the 16 element of Mueller matrix for a microsphere solution. The spheres diameter was 2μm and index of refraction 1.59. The Spheres were in a water media with index of refraction equal to 1.33. Each image is 4x4 cm. Light was incident at an angle θ from the normal.

The results for the meridian plane Monte Carlo are shown in Figure 3.19; identical results were achieved with the other 2 methods. The direction cosines at launch were changed to obtain an asymmetrical incident angle of θ. The programs reproduced all the shapes correctly, apart for the glare due to the air/glass/liquid interface, since I used matched boundaries in the program.

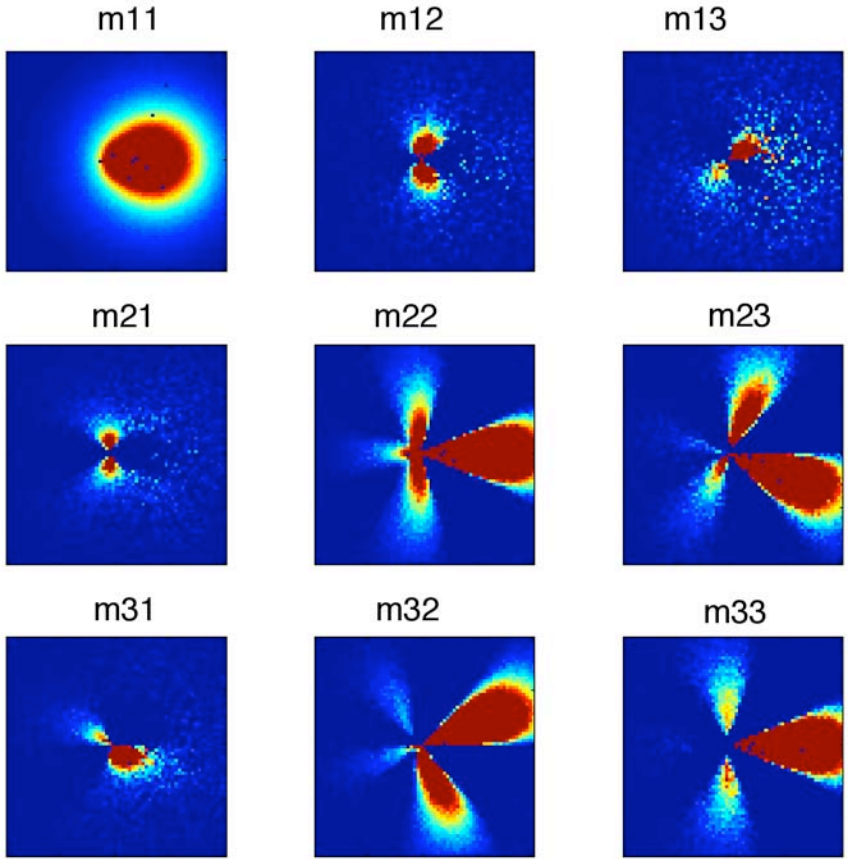

Fig. 3.19 - Monte Carlo simulation of the first 9 of elements of the 16 element of Mueller matrix. Illumination was asymmetrical.

3.8. Conclusions

In this chapter three different Monte Carlo programs that keep track of the status of polarization after every scattering event were described. The three programs are significantly different from each other yet they yield the same numerical results.

The programs were developed for Unix platform in C programming language. All three programs run at approximately the same speed. The three codes are available in appendix 2.

The Euler's angle Monte Carlo is easy to implement and intuitive, with a triplet of unit vectors following the photon reference frame after every scattering event. Euler's rotational matrices are well known since they are often used in rigid body rotations. Moreover since for any scattering event only one rotation of the reference frame is necessary the program should in principle be faster that the meridian plane method.

The quaternion Monte Carlo is an elegant solution to the Gimbal-Lock issue that afflicts the Euler's Monte Carlo. Quaternion rotations are used more and more often in computer science to handle animations. Many improvements are available to simplify the generation of the quaternion plus the multiplicative step in order to make them faster and I intend to explore these algorithms in the future.

The implementation of a robust Monte Carlo program was very important to our work because it allowed us to have a "gold standard" of polarized light scattering through simple media, such as microspheres solutions. With the Monte Carlo we were able to explore mechanism of scattering such as the importance of the size parameter or the scattering anisotropy in polarized light scattering. Some of these findings will be shown in the next chapter.

Chapter 4
Heuristic models of polarized light travel into scattering media

4.1 Introduction

As polarized light propagates through light-scattering media such as biological tissues, microsphere solutions, or an atmosphere with particulates, the polarization status of the light changes. The propagation of polarized light through a biological tissue causes the polarization status of the light to change due to tissue birefringence and tissue scattering. Imaging using polarized light can select light that is primarily backscattered from the superficial tissues. This is in contrast to light that reflects from the air/tissue surface, or to light that propagates deep into the tissue, and whose polarization status has been fully randomized before escape. Simple manipulation of polarized light images can be used to characterize the superficial region of the tissue, which is often the region where cancer develops [86].

Various groups have studied the transmission of polarized light through slabs of scattering media and made observations on how light depolarizes in this process. Jarry *et al.* [36] studied the randomization of linearly polarized light as it propagated through tissues and microsphere solutions, observing a surprising persistence of polarization

when propagating through liver tissues despite multiple scattering. Jacques *et al.* [37] reported similar findings and expanded the observation to three tissue types, skin liver and muscle. They proposed a heuristic model that described how the angle of orientation of linearly polarized light diffuses in angle space as the light propagates through tissue.

This diffusion process is characterized by an angular diffusivity, [rad^2/mean free path], that characterizes each tissue type and appears to scale with the birefringence of the tissue (skin > muscle > liver). Sankaran [87] studied the influence of densely packed microsphere solutions on the transmitted polarized beam, demonstrating that the behavior of polarized light in tissues is similar to the behavior in densely packed microsphere solutions. This result does not agree with the predictions of Mie theory for isolated microspheres. Studinski and Vitkin [88] proposed a method to examine polarized light interaction with tissues in the exact backscattered direction. They found that in the backscattered direction a significant fraction of the incident polarization is preserved, even for highly concentrated media and for biological tissues, this confirms theoretical findings of other authors as cited by Brosseau [19].

The wavelength spectrum of backscattered polarized light has been used to characterize the size of cellular particles such as the nuclei of cells. Backman *et al.* [25] used polarized light scattering spectroscopy to quantitatively measure epithelial cellular structures *in situ*. Perelman *et al.* [38] measured size distribution of nuclei of mucosal tissue using an optical fiber probe and polarized light. Sokolov *et al.* [40] reported a simple model that simulated the wavelength dependence of polarized light scattering from monolayers of microspheres and cells.

Ramella-Roman and Jacques [89] experimentally calculated the single scattering Mueller matrix for three biological tissues, liver, muscle, and skin. Their results showed that the single scattering Mueller matrix of highly birefringent materials such as skin and muscle could be approximated by the Mueller matrix for a simple retarder. Wang *et al.* [47] incorporated this idea into a Monte Carlo simulation composed of Mie scatterers embedded within a homogeneous birefringent medium.

In this chapter, a Monte Carlo simulation was used to model the experiments of Jacques *et al.* [37] in microsphere solutions. The Jacques heuristic model was used to determine the angular diffusivity, χ [radians^2/mfp] (mfp = mean free path between scattering events), that characterizes the randomization of polarization in the simulated experiments. These χ values obtained with the Monte Carlo data fitting and the c values obtained with the experimental fitting are compared and agree closely. The conclusion is that the heuristic model offers a convenient method for predicting the behavior of polarized light in light-scattering media such as tissue. The heuristic model can predict the behavior of polarized light far distant from a source which can be very time consuming when using Monte Carlo simulations.

4.2 Development of an heuristic model - the first χ model

Consider a tissue composed of scattering particles and/or small domains of local birefringence. Each small birefringent domain has its fast and slow axes randomly oriented. An example would be the collagen fibers of the papillary and reticular dermis of skin. Even a tissue such as muscle that is grossly composed of oriented fibers has microdomains of birefringent orientation that are independently oriented [80]. When linearly polarized light is scattered, the orientation of its linear polarization can change. As linearly polarized photon propagates through a birefringent medium, the de-phasing of its parallel and perpendicular field components causes its polarization to transform from linear to circular polarization, then returning to linear. Deflection due to scattering can frustrate this orderly cycle of polarization. In summary, a rather complex process of linear polarization loss occurs as the light propagate through a tissue.

For an infinitesimal path length of propagation, the ensemble of photons comprising the beam is expected to undergo a net incremental loss of polarization. The question is whether a simple heuristic model for de-polarization can usefully approximate this complex process. One simple model assumes that the orientation of linearly polarized light migrates either clockwise or counter-clockwise according to a diffusion

process. As the distribution of polarization orientation broadens, the ensemble of states approaches a completely de-polarized state. The model ignores the true nature of how birefringent tissue would de-phase linear polarization. Yet, such a simplified model successfully mimicked the process of de-polarization in microsphere solutions and tissues [37]. Therefore, could the heuristic model also mimic the process of loss of linear polarization demonstrated in a Monte Carlo simulation?

A simple transmission mode set-up is shown in Figure 4.1. When the analyzer is oriented parallel to the linear polarization of the incident light, the collected power is called I_{par} [W].

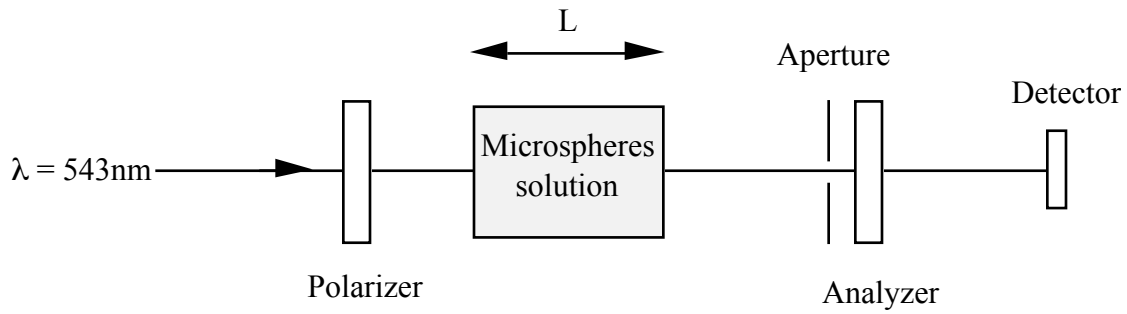

Fig. 4.1 – Simple transmission mode set-up that we can model with the heuristic model. The analyzer can be rotated in two positions: co-polarized and cross-polarized.

When oriented perpendicular to the incident light, the collected power is called I_{per} [W]. The I_{par} measurement collects both the unscattered original beam and a portion (f) of the scattered light that is oriented parallel to the incident light.

$$I_{par} = I_o \exp(-\mu_s L) + f\, I_{par\ scattered} \qquad (4.1)$$

The portion of the beam that travels through the media without scattering equals $I_o \exp(-\mu_s L)$, where μ_s [cm^{-1}] is the scattering coefficient of the medium, and L [cm] is the thickness of the slab. The portion of the beam that is scattered is $1 - I_o \exp(-\mu_s L)$.

The I_{per} measurement collects a portion of the scattered light that is oriented perpendicular to the incident light

$$I_{per} = f\, I_{per\ scattered} \tag{4.2}$$

After propagating through some optical thickness τ (equals $\mu_S L$), the probability for the orientation of polarized light will be at an angle ϕ relative to its initial orientation is approximated by the Gaussian expression [37]

$$p(\phi) = \frac{\exp(-\frac{\phi^2}{2\sigma^2})}{2\sigma\sqrt{\pi/2}} \tag{4.3}$$

where the value χ is equated to the product of the optical thickness τ [mean free paths or mfp] and an angular diffusivity χ [radians^2/mfp].

$$2\sigma^2 = \chi\tau$$

and

$$\int_{-\infty}^{\infty} p(\phi)\ d\phi = 1$$

The angle ϕ can diffuse in both the $+\phi$ and $-\phi$ directions relative to the original parallel orientation defined as $\phi = 0°$. While ϕ can diffuse beyond an angle of $\pm 2\pi$, the factor $p(\phi)$ was originally considered as a Gaussian that vanished at large ϕ.

The amount of parallel light is proportional to the expectation value for $\cos^2(\phi)$, since detected power equals the square of the parallel component of the electric field, which is proportional to $\cos(\phi)$.

$$\int_0^{\infty} p(\phi)\cos^2\phi\, d\phi = \frac{1}{2}\left(1 + \exp(-2\sigma^2)\right) \tag{4.4}$$

$$\int_{-\infty}^{\infty} p(\phi)\cos^2\phi\, d\phi = \left(1 + \exp(-2\sigma^2)\right)$$

The amount of parallel light is proportional to <$\cos^2\phi$>:

$$\left\langle \cos^2\phi \right\rangle = \frac{\int_{-\infty}^{\infty} p(\phi)\cos^2\phi \, d\phi}{\int_{-\infty}^{\infty} p(\phi) d\phi} = \frac{1}{2}\left(1 + \exp(-2\sigma^2)\right) \tag{4.5}$$

$$\left\langle \cos^2\phi \right\rangle = \frac{\int_{-\infty}^{\infty} p(\phi)\cos^2\phi \, d\phi}{\int_{-\infty}^{\infty} p(\phi) d\phi} = \frac{1}{2}\left(1 + \exp(-2\sigma^2)\right) \tag{4.6}$$

Substituting $\chi\tau$ for σ^2, the following descriptions of $I_{par\ scattered}$ and $I_{per\ scattered}$ are expected

$$I_{par\ \ scattered} = I_o(1 - \exp(-\tau))\frac{1}{2}(1 + \exp(-\chi\tau)) \tag{4.7}$$

$$I_{per\ \ scattered} = I_o(1 - \exp(-\tau))\frac{1}{2}(1 - \exp(-\chi\tau))$$

The attenuation of scattered light due to absorption, scattering, and possible losses at lateral boundaries requires including an additional term $\exp(-\mu_{att}L)$.

$$I_{par\ \ scattered} = I_o(1 - \exp(-\tau))\exp(-\mu_{att}L)\frac{1}{2}(1 + \exp(-\chi\tau)) \tag{4.8}$$

$$I_{per\ \ scattered} = I_o(1 - \exp(-\tau))\exp(-\mu_{att}L)\frac{1}{2}(1 - \exp(-\chi\tau))$$

This factor accounts for the slow but steady loss of $I_{par\ scattered}$ and $I_{per\ scattered}$ at larger τ. The term μ_{att} was empirically specified from the data of each experiment or

simulation, but is related to the $\mu_{eff} = (3\mu_a(\mu_a + \mu_s'))^{1/2}$ of diffusion theory.

$$\frac{I_{par}}{I_o} = \exp(-\tau) + \frac{f}{2}(1-\exp(-\tau))\exp(-\mu_{att}L)(1+\exp(-\chi\tau)) + \beta \tag{4.9}$$

$$\frac{I_{per}}{I_o} = \frac{f}{2}(1-\exp(-\tau))\exp(-\mu_{att}L)(1-\exp(-\chi\tau)) + \beta$$

Where a final experimental parameter β has been added to account for non-perfect polarizers. In general β will not contribute significantly to I_{par} but will dominate the I_{per} measurements as τ goes to 0. For example for τ=0, there should be no perpendicular light reaching the detector, but depending on the polarizers used I_{per}/I_o will range from 10^{-5} to 10^{-4}. Finally, a factor called Pol is calculated using I_{par} and I_{per} from equation 4.8

$$Pol = \frac{I_{par} - I_{per}}{I_{par} + I_{per}} \tag{4.10}$$

In summary, the behavior of I_{par}, I_{per}, and Pol for light transmitted through complex tissues and exiting near to on-axis within a narrow on-axis solid angle of collection is function of the optical thickness τ and three fitting parameters χ, f, and μ_{att}.

4.3 Experimental results[1]

Using the experimental apparatus shown in Figure 4.2, a helium-neon laser light source was used, either a 543-nm or 633-nm wavelength, with a beam size of 1.5 mm diameter. The polarizing elements (Ealing Electrooptics, plc., Watford, UK) had an extinction equal to 1e-6.

A beam of linearly polarized light with power I_O [W] is incident on a slab of scattering medium.

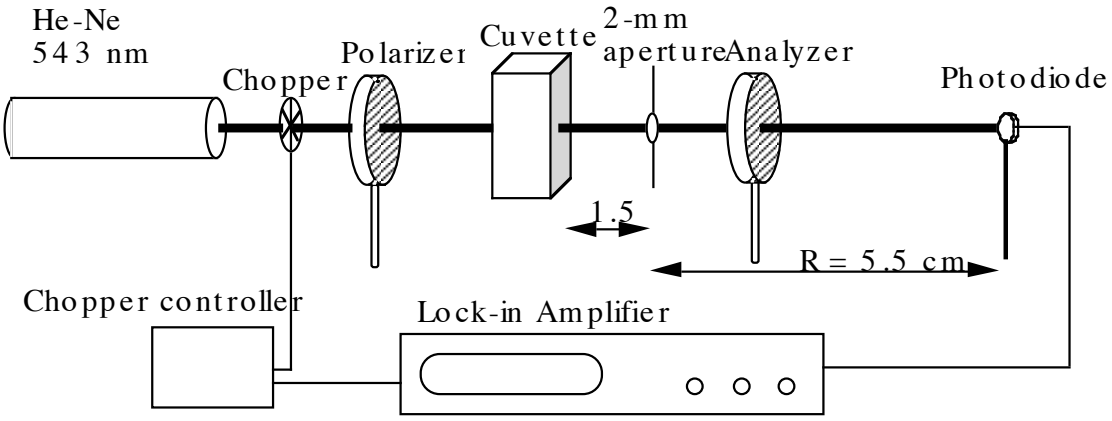

Fig. 4.2. - Transmission experiment. The second polarizer between the cuvette and detector is an analyzer to select transmitted light either parallel or perpendicular to the axis of the first polarizer.

An analyzing linear polarizer is placed in front of the detector. An aperture with diameter d_1 is located a distance R_1 from the rear of the slab, and a detector with diameter d_2 is located a distance R_2 from the slab. The choices of d_1, R_1 and R_2 are such that the solid angle of collection by the detector equals the solid angle of collection by the aperture (for a point source of diffuse excitance on axis at the slab's rear surface). For d_1 = 0.214 cm, d_2 = 1 cm, R_1 = 1.5 cm, and R_2 = 5.5 cm, the solid angles of collection by the aperture and the detector match and equal 0.016 sr. If the exittance of scattered light from this point source on the slab were Lambertian, the solid angle of collection would collect a fraction (0.005) of the total exittance. Transmitted light that is not lambertian,

[1] Parts of this section where taken from: Jacques SL, Roman R J, Lee K Imaging "Superficial Tissues With Polarized Light" Lasers Surg. Med. 26:119–129, 2000.

(e.g. forward scattered) will change the fraction of light collected (e.g. increase f).

Experiments were conducted on mono-disperse solutions of microspheres and on three biological tissues (liver, muscle, skin). The liquid samples were held in an adjustable cuvette, with front and rear plastic windows. The sides of the cuvette were black. The path length through the cuvette was adjusted from 0.1 to 1.5 cm to yield a range of optical depths.

In the tissue experiments, thin sections of tissue were cut to desired thicknesses, and then sandwiched between the windows of the cuvette with slight pressure to enforce a known thickness. In some cases, such as chicken liver and muscle, the tissue thickness (smallest obtainable thickness was 100μm, largest thickness was 3mm) could be varied by progressively squeezing the tissue between glass slides. For the pig tissues, which were firmer than chicken tissues, the samples had to be cut to approximately the desired thickness before sandwiching between the glass

Figures 4.3 and 4.4 show some typical examples for spheres and chicken liver at 633 nm.

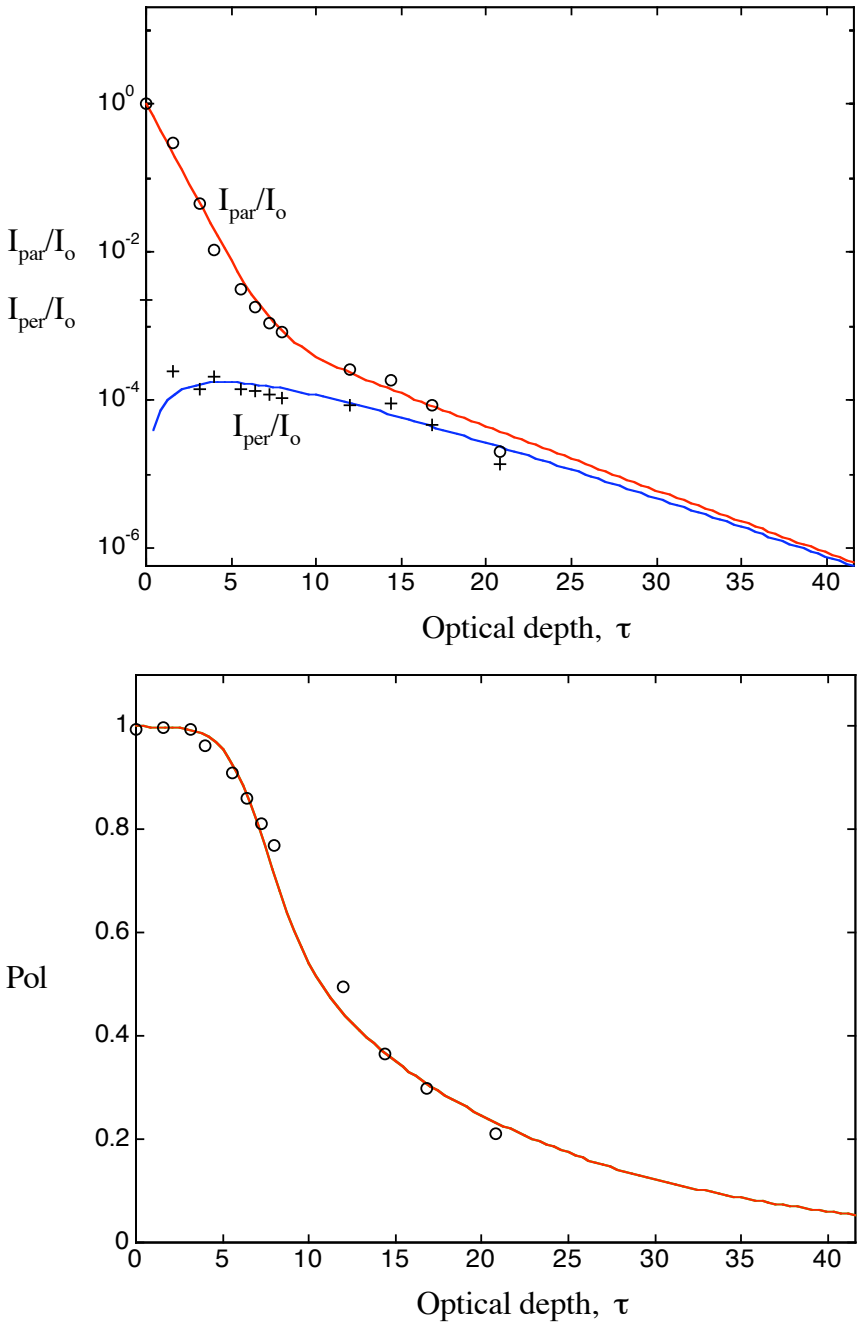

Fig. 4.3. – Experimental data for polystyrene microspheres in water. Sphere diameter was 482nm, μ_s=8cm^{-1}. Wavelength was 543nm, Solid lines are best fit of equation 4.8 and 4.9 (below) with f = 0.005, χ = 0.8, μ_{att} = 1.5 $\beta = 10^{-4}$.

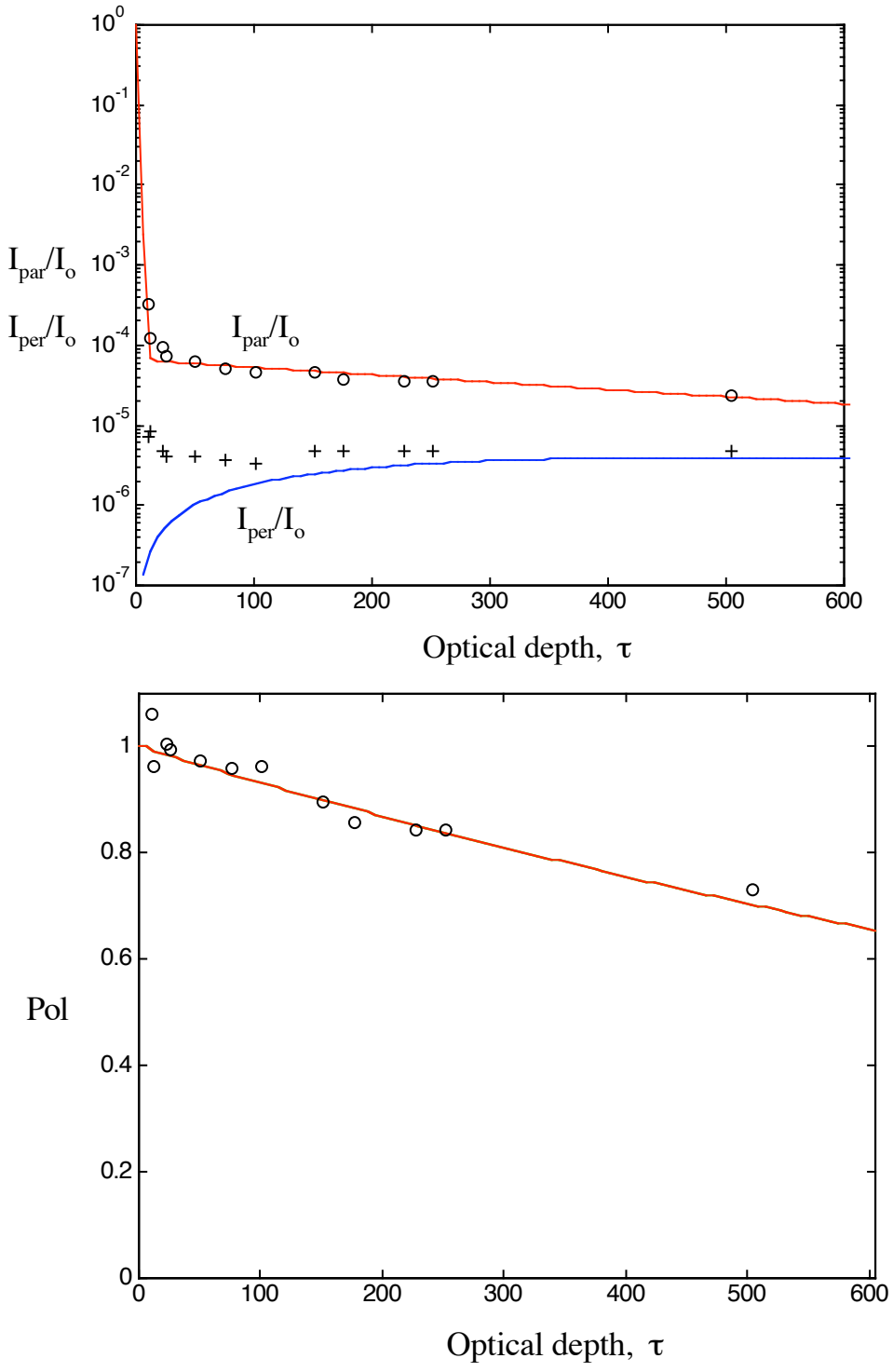

Fig. 4.4. – Experimental results for chicken liver. Lines are best fit to data with $f = 0.65\ 10^{-4}$, $\mu_{att} = 0.45$ cm^{-1}, and $\chi = 0.0007$ $\beta = 10^{-5}$.

The two examples above are a polystyrene microsphere solution in Figure 4.3 and chicken liver in Figure 4.4. The microspheres had a diameter of 482 nm and were at a concentration of 0.0026% volume fraction or, C = $4.52 \bullet 10^{8}$ cm^{-3}, which yielded a scattering coefficient of μ_s= 8.0 cm^{-1} based on Mie theory for spheres with an index of refraction n= 1.59 (water n=1.33). Fitting the data, the values f = $30 \bullet 10^{-4}$, μ_{att} = 1.5 cm^{-1} , and χ = 0.070 rad^2/mfp were obtained. For the chicken liver sample the scattering coefficient was estimated from the values of reported μ_s values for liver in the table of optical properties compiled by Cheong *et al.* [91] to be about μ_s = 290 cm^{-1} at 633-nm wavelength. Fitting the data, the values f = 0.65 10^{-4}, μ_{att} = 0.45 cm^{-1} , and χ = 0.0007 rad2/mfp were obtained. Other microsphere solutions and tissue samples showed similar behavior. Figure 4.4 show a summary of the χ values obtained from the analyses based on the mean three samples with standard deviations in the range of 10–50% of the mean.

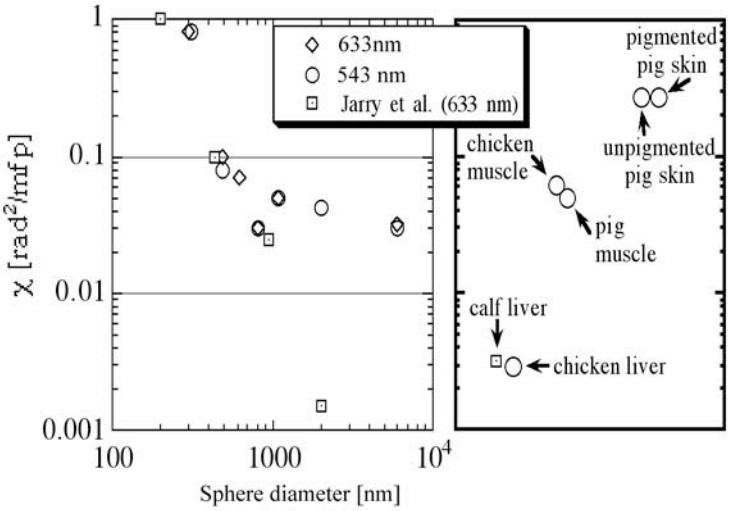

Fig. 4.5. - Summary of diffusivity χ [rad^2/mfp] of polystyrene microspheres and various *ex vivo* tissues. (Left) Diffusivity plotted versus microsphere diameter. (Right) Tissue diffusivity. Data from this study are represented as circles and diamonds. Data from Jarry *et al.* [17] are represented as squares.

On the left, the x-axis is the microsphere diameter (nm) and the y axis is the χ value (rad^2/mfp). On the right, the χ values for tissues are plotted in order of increasing χ value. The results show data at both the 543 and 633 nm wavelengths. Figure 4.5 also

shows the χ values derived from the experimental data of Jarry *et al.* [36] for microspheres and calf liver. The analysis for their calf liver assumed the same scattering as above, μ_s = 290 cm^{-1}. The domestic and Yucatan pig skin samples yielded nearly identical χ values, so the melanin content did not affect the measurement. The value of χ varied in the order, liver < muscle < skin.

4.4 Analysis of χ model

The hypothesis of de-phasing of the polarization following every scattering event, first introduced in section 4.2, was studied with Monte Carlo simulations. The analysis were performed with the meridian plane Monte Carlo described in chapter 3.

4.4.1 Modeling with Monte Carlo of the experiment, revisiting the heuristic fit.

The first test was validated the Monte Carlo program correctly using experimental results for microspheres. In Figure 4.6 and 4.7 I show experimental data for a mono-disperse solution of microspheres of diameters 482 nm and 308 nm, and the corresponding results from three Monte Carlo runs with (different random number seeds). For both experimental data and simulations the source was a parallel polarized beam of 543 nm wavelength. In the Monte Carlo simulations a factor β due to the extinction of the polarizers, measured at $2 \cdot 10^5$, was added.

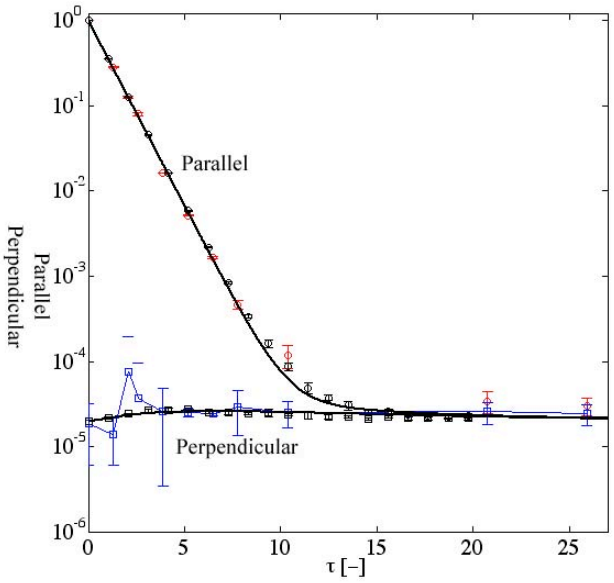

Fig. 4.6 – Experimental results and Monte Carlo simulations (black circles and squares) for a monodisperse solution of microspheres. Diameter of microspheres was 482 nm, light wavelength was 543 nm. The back line is the heuristic model applied to the Monte Carlo data $\chi = 0.1$. The dashed line is the original prediction based only on the experimental data χ=0.07, f=3•10^{-4}, μ_{att}=1.0. The experimental extinction of the polarizer analyzer pair was 10^{-5}.

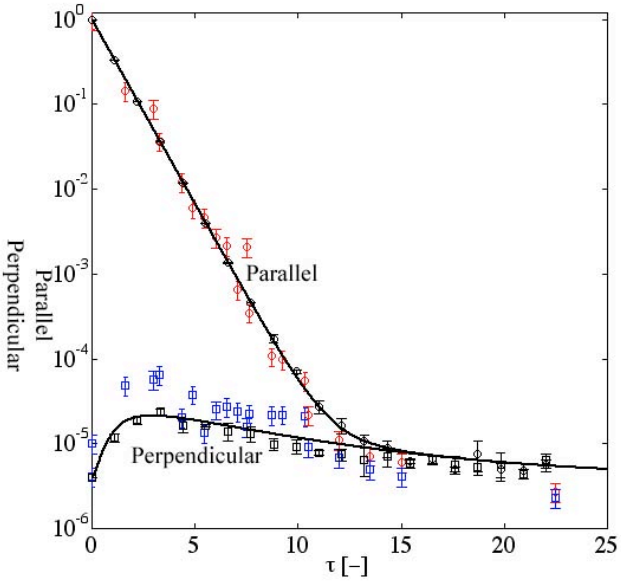

Fig. 4.7 – Experimental results and Monte Carlo simulations (black circles and squares) for a monodisperse solution of microspheres. Diameter was 0.308 μm, light wavelength was 0.543 μm.

The heuristic fit proposed in the previous paragraph was applied to the data and is visible as a dashed black line in both Figure 4.6 and 4.7.

4.4.2 Influence f and χ in the heuristic model.

The fitting parameters χ, f, and μ_{att} are critical in this model. A simple collection geometry is modeled to evaluate the limit of the heuristic model. A parallel polarized pencil beam of wavelength 543 nm penetrates through a slab of microspheres solution of increasing optical thickness τ. Small spheres of radius 0.01μm are considered. An aperture of variable diameter d is located at a distance R = 7 cm from the slab. This geometry, shown in Figure 4.8, was modeled with Monte Carlo simulations. The results of the simulations and the corresponding heuristic model are shown in Figure 4.9

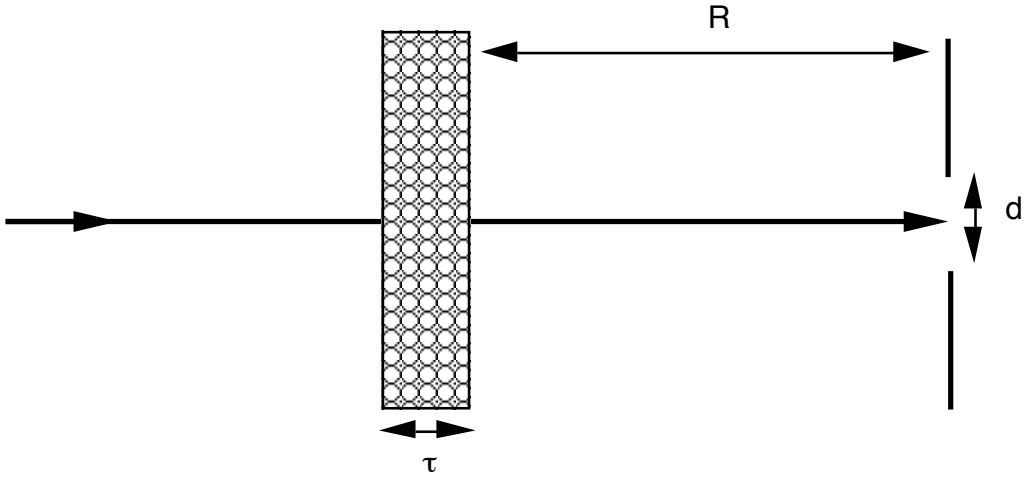

Fig. 4.8 - Pencil beam (λ = 543 nm) impinging on a slab of variable optical thickness τ, a detector behind an aperture of radius a collects a fraction of the transmitted light. Three apertures of increasing diameter d are studied 0.2 cm, 2 cm and 4 cm, the detector is located at a distance R from the slab.

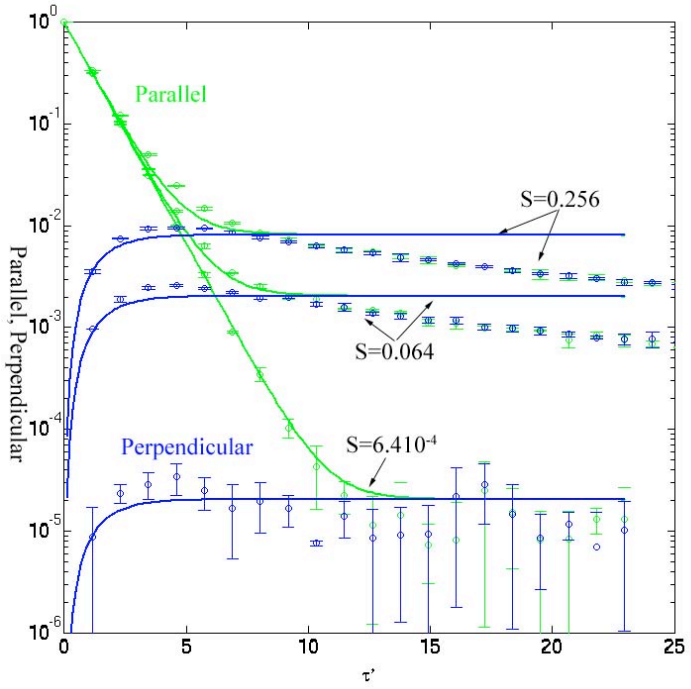

Fig. 4.9 – Monte Carlo simulations (circles) and heuristic model fits for the slab geometry of Figure 4.8. Parallel and perpendicular results are modeled with the heuristic model. Model fits parameters are given in table 4.1

The heuristic model needs the attenuation parameter μ_{att} to yield good fits. The μ_{att} value is related to μ_{eff} but it is also related to the particle size and χ. Adding μ_{att}=0.95 Figure 4.11 is obtained for the three aperture sizes. The parameters used in these fits are shown in table 1.

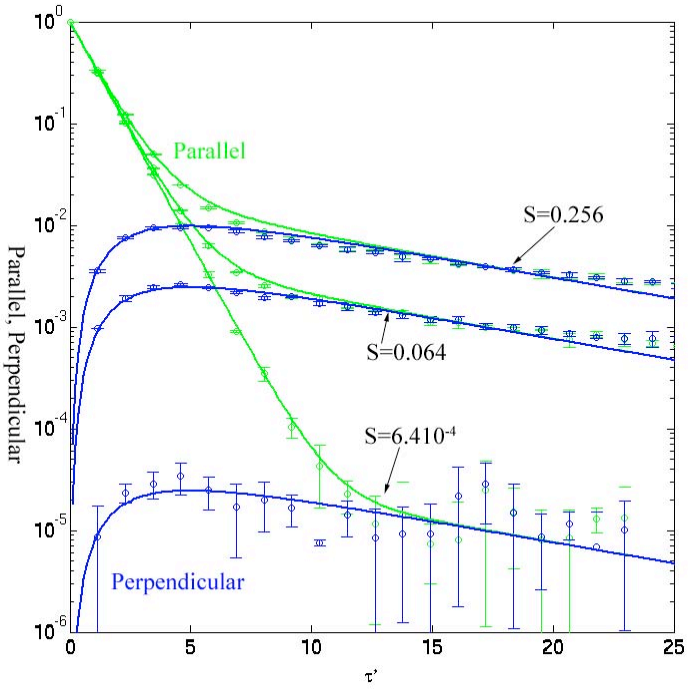

Fig 4.10 – The heuristic model fits much better when the μ_{att} parameter is included. Fits results are tabulated in Table 1.

Aperture d [cm]	Solid angle (S) [sr]	β [−]	μ_{att} [-]	f [-]	χ [rad²/mfp]
0.2	6.410^{-4}	0	0.095	1.0210^{-4}	0.3
2.0	0.064	0	0.095	0.0120	0.3
4.0	0.256	0	0.095	0.0408	0.3

Table 4.1 – Fitting parameters used in the fits of Figure 4.10 and 4.11. In Figure 4.10 μ_{att}=0.

4.4.3 The χ parameter and the de-phasing of the polarization for increasing τ.

The original hypothesis behind equation 4.8 was that polarization was rotated of an angle ϕ at each scattering event. This rotation of de-phasing was assumed to be Gaussian. In the meridian plane Monte Carlo program, the polarization reference frame is rotated about the direction of propagation by an angle ψ. This is followed by scattering at an angle θ, and finally rotating the reference frame back to the meridian plane. If ϕ is defined as the angle needed to rotate the polarization frame to that of the detector, then it is desirable to observe how ϕ changes with increasing number of scattering events.

Using the a semi-infinite geometry of Figure the meridian plane Monte Carlo code was modified to collect the value of ϕ for every transmitted photon as well as the number of scattering steps for that photon. A mono-disperse solution of 1.0 μm microspheres was used, the scattering coefficient μ_s was 11 cm^{-1}. The results are shown in Figure 4.11. The histograms of the distribution of ϕ angle for 1, 6, 8, and 10 scattering events are shown. The distribution was normalized by the value at ϕ at 0, which is the number of photons whose polarization was not affected. Apart from the peak at 0, ϕ is uniformly distributed between –180 and 180. This is to be expected since the angle ψ in the rejection method is nearly uniformly distributed. After 5 scattering events, more and more photons have changed reference frame and ϕ values different from 0 start being visible. These values become more and more visible after 6 scattering events. At 10 scattering events there is no difference between the scattered and unscattered photons.

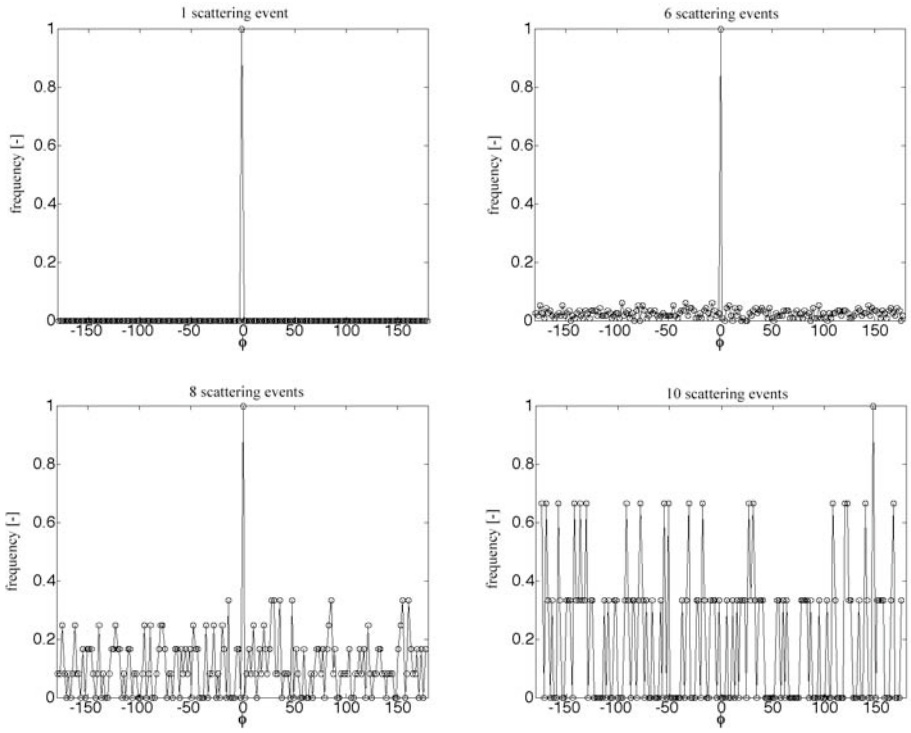

Fig. 4.11 – Rotation angle ϕ for traveling through a mono-disperse solution of microspheres of 1μm diameter. The distribution of the angle ϕ is plotted for 1, 6, 8, and 10 scattering event.

4.4.4 Poincaré sphere analysis

The French mathematician Henri Poincaré [92] used a sphere to represent the change in the status of polarization of a polarized light beam that interacts with any optical device. The Poincaré sphere is useful and intuitive since every point on the sphere corresponds to a possible status of polarization. The Poincaré sphere is shown in Figure 4.12. The 3 coordinates of a point on the sphere [Q, U, V] are the last three elements of the Stokes vector. The radius of the sphere is the first element of the Stokes vector I. The angle ϕ_i is called the azimuth angle and the angle ε is called the ellipticity angle.

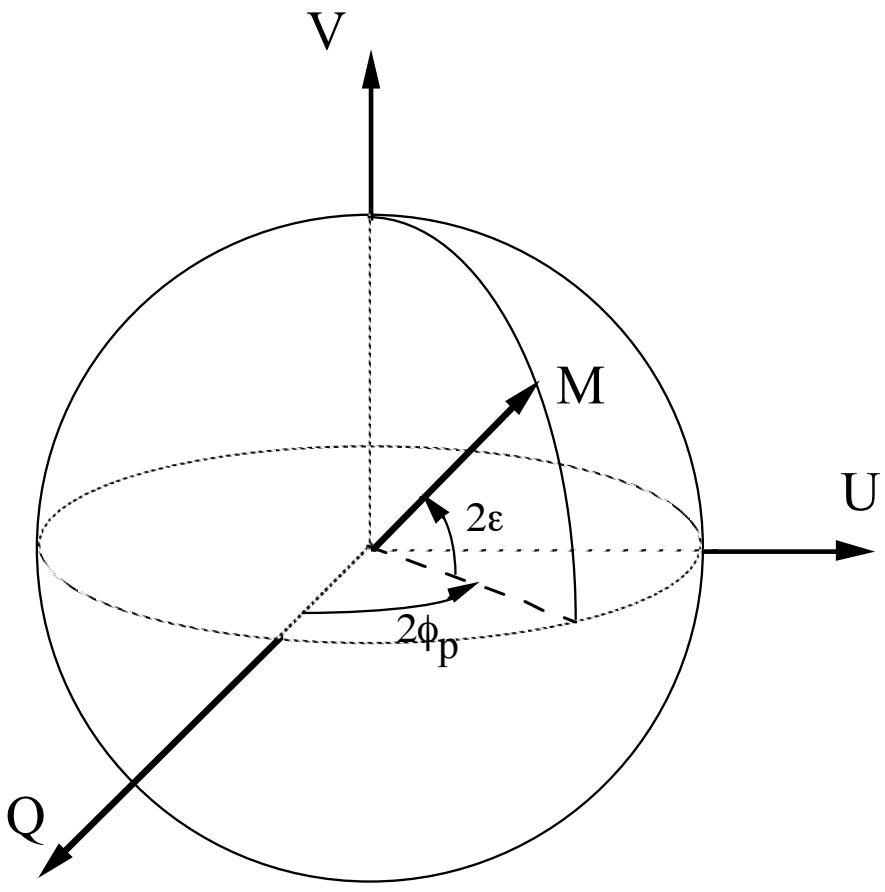

Fig. 4.12 The Poincaré sphere. The coordinates of M are the last three elements of the Stokes vector, [Q, U, V]. The radius of the sphere is the first element of the Stokes vector I.

The relation between these angles and the Stokes vector is shown in equation 4.11.

$$
\begin{aligned}
I &= E_x^2 + E_y^2 \\
Q &= E_x^2 - E_y^2 = I\cos(2\varepsilon)\cos(2\phi) \\
U &= 2E_xE_y\cos(\Gamma) = I\cos(2\varepsilon)\sin(2\phi) \\
V &= 2E_xE_y\sin(\Gamma) = I\sin(2\varepsilon)
\end{aligned}
\tag{4.11}
$$

The Poincaré sphere model was used to depict the fate of a polarized light beam as it undergoes multiple scattering events. A horizontally polarized light beam penetrating a semi-infinite media made of latex microspheres was simulated using meridian plane Monte carlo. The Stokes vector was collected, and the ε and ϕ_p angles were calculated after a set number of scattering events. Histogram of all the values of ε and ϕ_p for a specific number of scattering is sown in Figure 4.13 and 4.14.

The red line is the outline of the histogram for 2 scattering events and the green line is the outline for the histogram for 19 scattering events. The curves were normalized to their value at 0˚. The ellipticity of the polarized beam increased in width after every scattering event. The ellipticity distribution behave differently for very small sized spheres (10nm), rather than larger spheres (300nm). No significant difference in ellipticity is visible between large spheres (300nm, 2000nm, and 10000m).

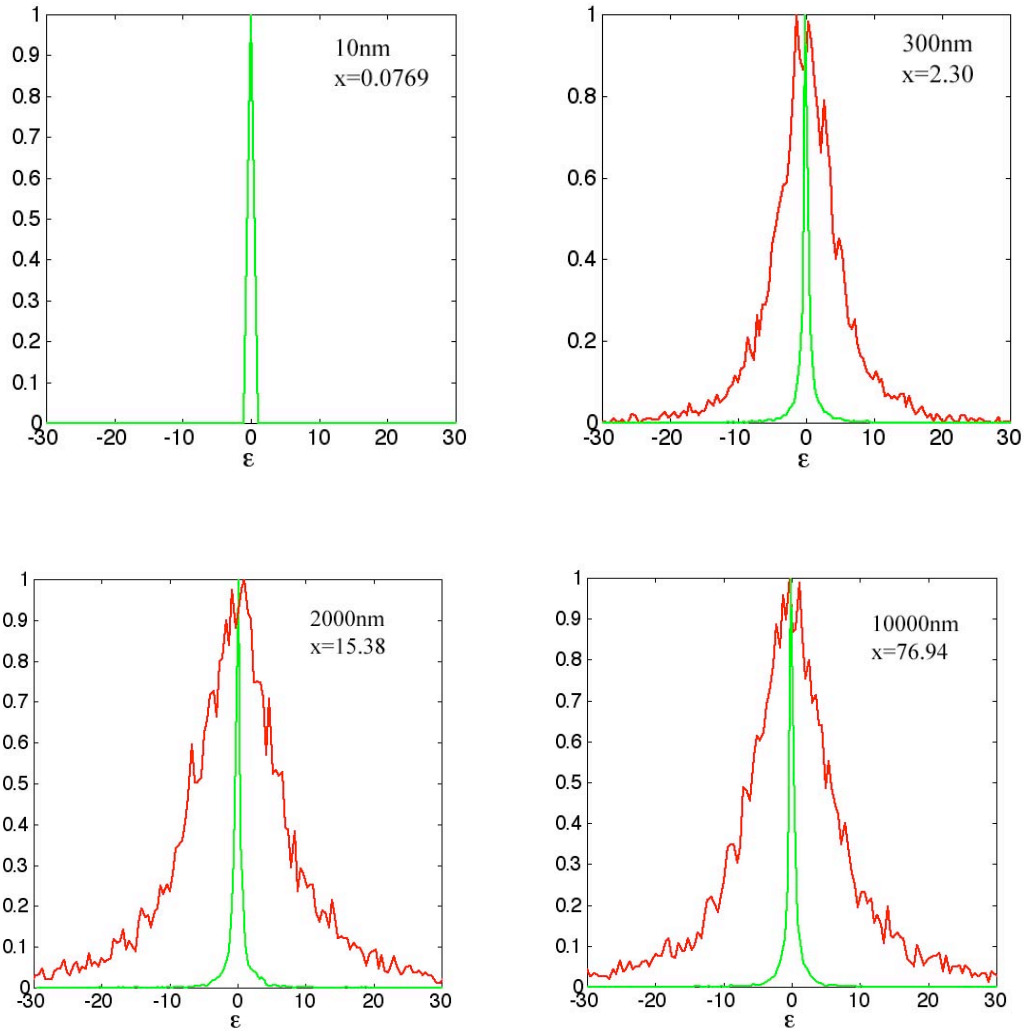

Fig. 4.13 - Ellipticity for 4 sphere sizes (0.01 μm, 0.3μm 2μm and 10μm) the red line is the distribution of ellipticity for 2 scattering events, the green line is the ellipticity of 19 scattering events, the lines in between are the ellipticity distribution fro 3 to 18 scattering events.

A more significant dependence on the particle size is visible in the azimuth angle ϕ. In Figure 4.14 the distribution of ϕ for 0.01 μm, 0.3μm, 2μm, and 10μm is shown. The red line corresponds to the outline of the histogram of ϕ for 2 scattering events. The green line corresponds to the outline of the histogram of ϕ for 19 scattering events. It may be noticed that the distribution of ϕ that is Gaussian after 2 scattering events broadens to a flat top for 0.01 μm and 0.300μm at 19 scattering events. For larger

spheres, 2μm and 10μm, the Gaussian shape is still maintained although it is broader after 19 scattering events.

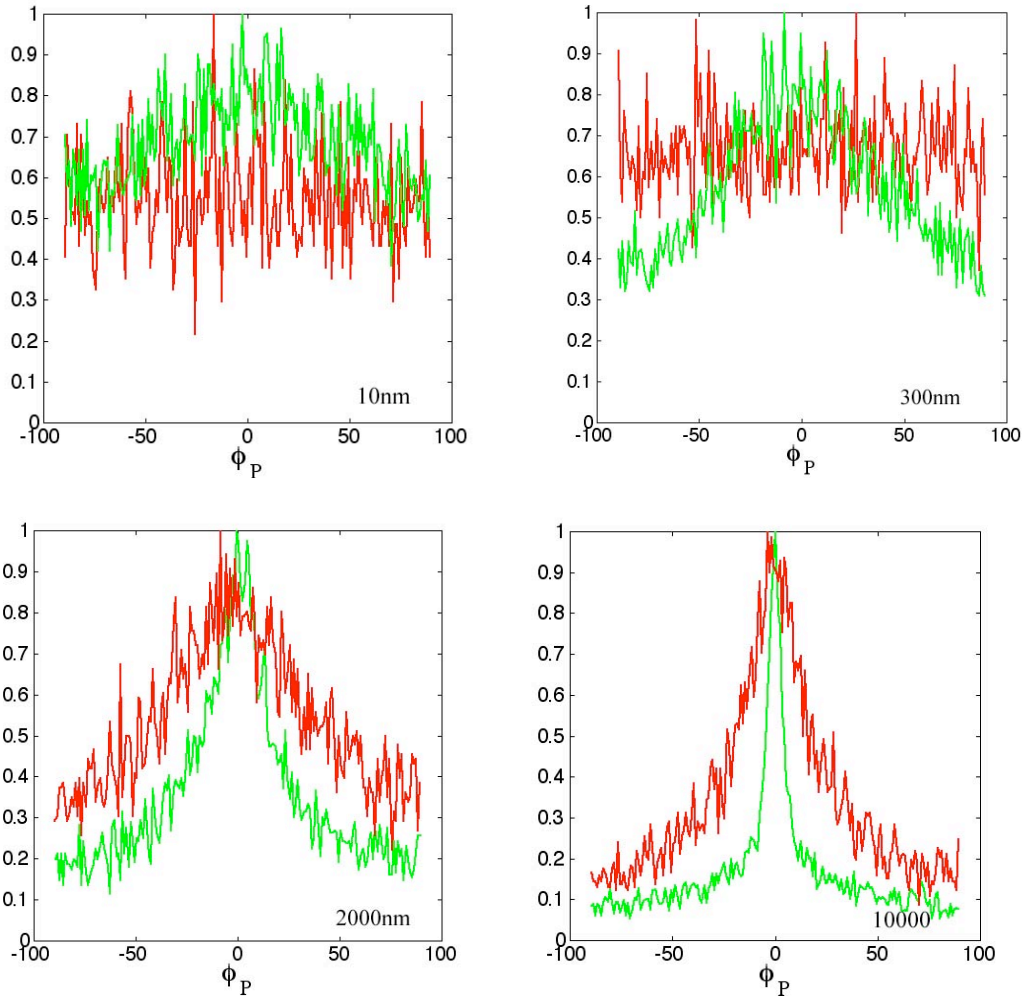

Fig. 4.14 - Azimuthal angle frequency distribution for 4 sphere sizes. The green line is the distribution of azimuthal angle for 2 scattering events, the red line is the azimuth angle after 19 scattering events.

Projecting the values of ϕ and ε on a complex plane an idea how the polarization of light distributed itself after the scattering events may be obtained. The values of ϕ_P and ε are used to calculate the complex number ξ.

$$\xi = \frac{\tan(\phi) + i\tan(\varepsilon)}{1 - i\tan(\varepsilon)\tan(\phi)} \tag{4.12}$$

In the complex plane all the values on the imaginary axis correspond to elliptically polarized light the values. Im(ξ) = ± i corresponds to right and left circular polarized light, Figure 5.15. All the values on the real plane correspond to linearly polarized light states. At the origin light is horizontally linearly polarized and changes to perpendicular polarized light for points approaching ± ∞.

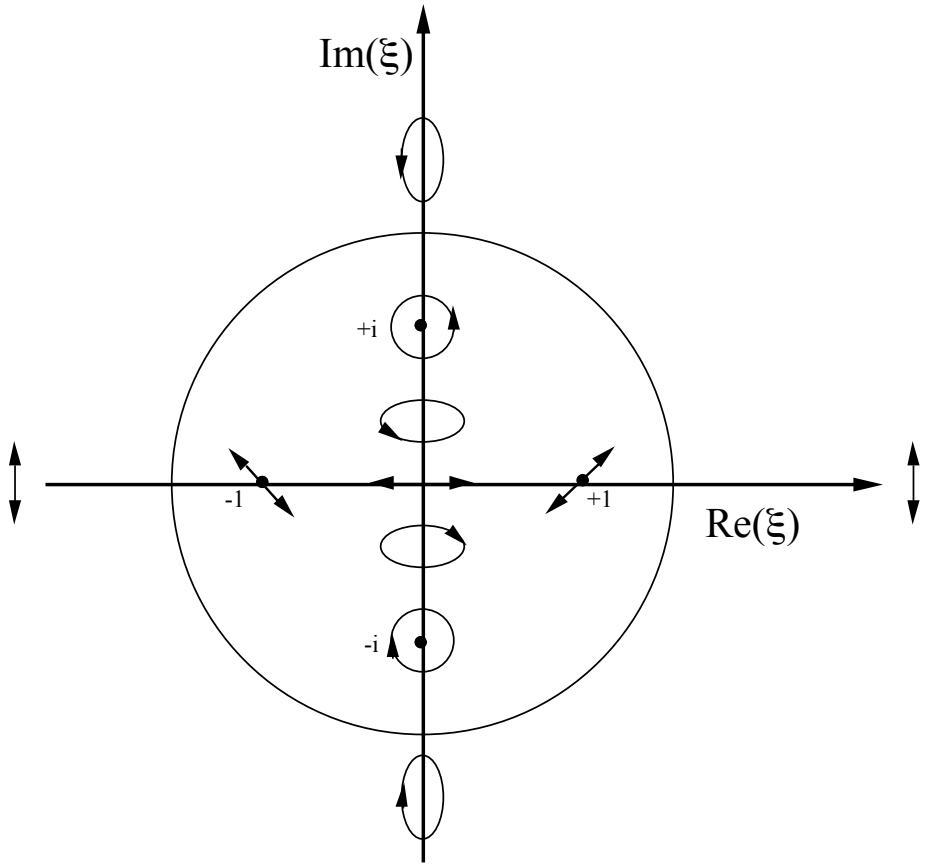

Fig. 4.15 - Representation of the states of polarization in the complex plane

The values of ϕ_P and ε used in the previous simulations are used to calculate ξ. The results are visible in Figure 4.17. When the scatterers are very small (10nm) almost no light goes into an elliptical state; light becomes more and more perpendicularly linearly polarized. For larger sizes, more elliptical light is visible.

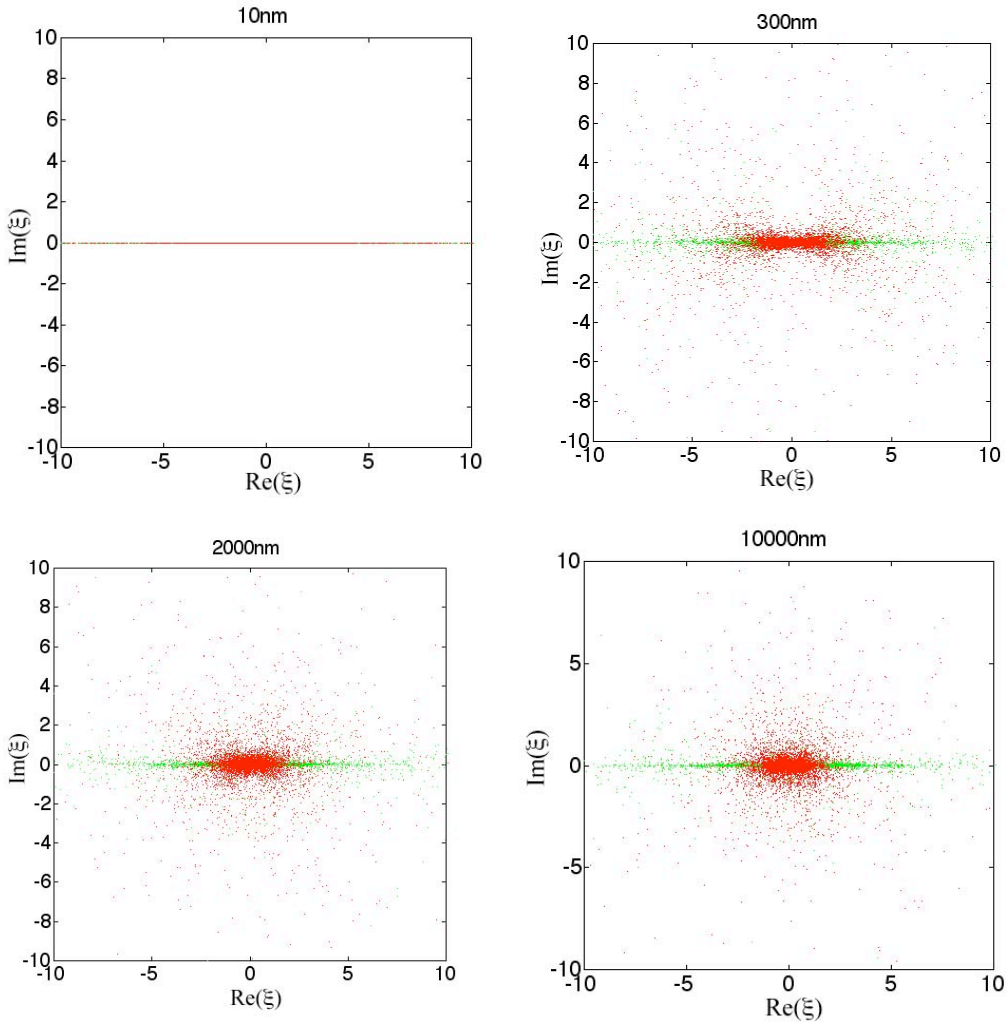

Fig. 4.16 - Complex representation of the status of polarization of the photon after 2 and 19 scattering events. Green data is for 2 scattering events red data is 19 scattering events.

This representation suggests that smaller spheres depolarize differently than larger spheres. The depolarization for smaller spheres happens very quickly but the polarization migrates towards other linear states, no circularly polarized light is created. For larger spheres the migration from linear states of polarization to elliptical states is more visible. This suggests that a parameter that accounts for the generation of elliptical states might be added to the present model that considers only the linear states. Nevertheless, the hypothesis of a Gaussian distribution of a de-phasing angle seem to be appropriate for the model.

4.4.5 Polarized light distribution through a slab

Brosseau [19] showed experimentally and with Monte Carlo models that the degree of polarization of a polarized beam passing through a slab is affected by an increase in entropy caused by scattering events. The meridian plane Monte Carlo was adapted for a a slab geometry shown in Figure 4.17. Two dimensional maps of polarized light scattering were generated for various optical thickness. Three different sphere sizes were considered: 10 nm, 482nm, and 2000nm. The index of refraction of the spheres were 1.59 and the index of refraction of the surrounding media was 1.33. The beam wavelength was 543nm and the beam polarization was parallel at launch (Stokes vector =[1 1 0 0]). Three maps were generated using a detector analyzer: parallel, perpendicular and degree of polarization.

One hundred optical thickness were tested ranging from 0 to 1, this resulted in 100 thicknesses z ranging from 0 to 1cm.

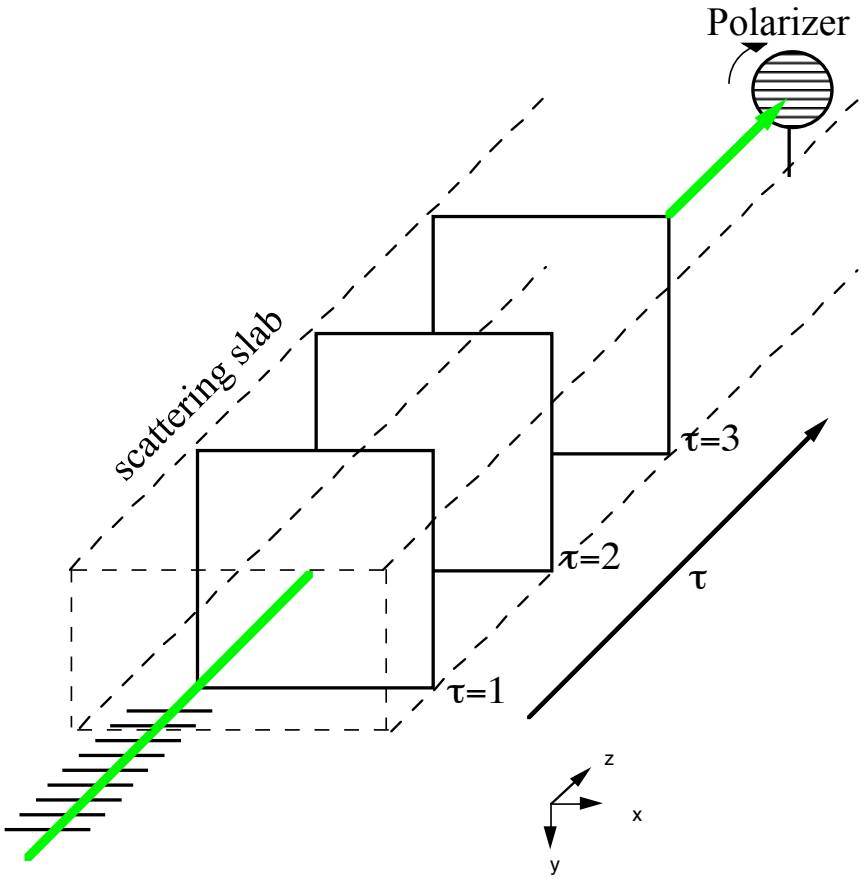

Fig. 4.17. – Geometry of Monte Carlo runs. Two-dimensional maps of polarized light scattering are generated for increasing t.

The images in Figure 4.18, 4.19, and 4.20 were obtained by stacking the 2D images obtained for each simulation thickness. The Figure show slices through two directions: across the plane xz and yz. As expected, smaller spheres de-polarize light faster than larger spheres as is evident when comparing the bottom image of Figure 4.18 and 4.20, near x=0, y=0. At thickness of z =0.5 cm in Figure 4.18 the degree of polarization has decreased to 0.4 while in Figure 4.20 at the same thickness the degree of polarization is 0.9. A characteristic glow-ball shape is visible for all three spheres sizes, yet it is evident as the perpendicular spread is much higher for smaller spheres, consistent with the particles phase function structures.

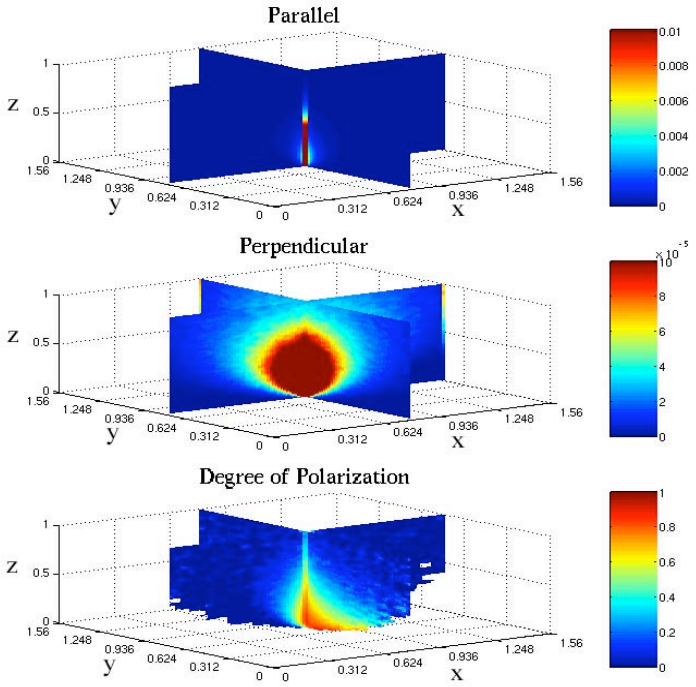

Fig. 4.18. – 10nm spheres. XZ and YZ sections are showed in all graphs. The axis are in cm. Parallel (top image), the crossed transmission (middel image), and the degree of linear polarization of the transmitted light (bottom image).

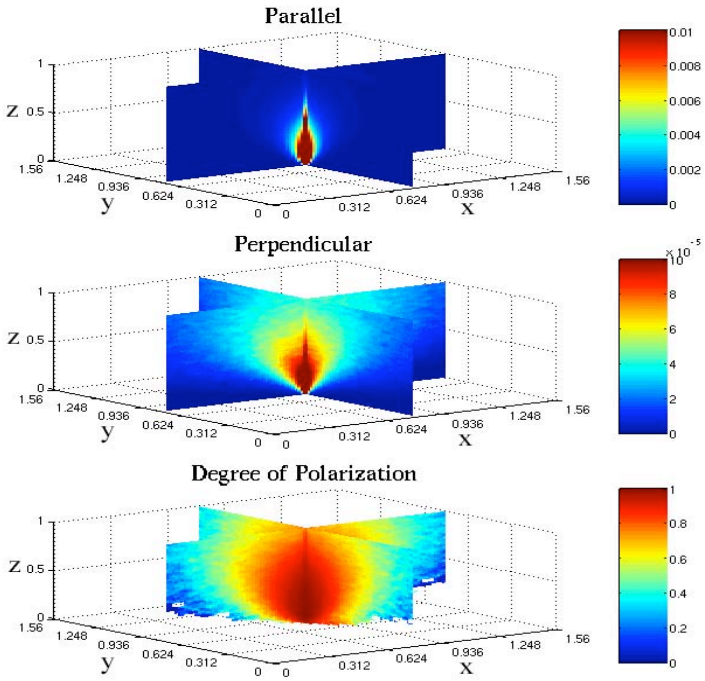

Fig. 4.19. – Transmission through various thickness z, showing x-y variation from 482nm. μ_s=11 cm^{-1}.

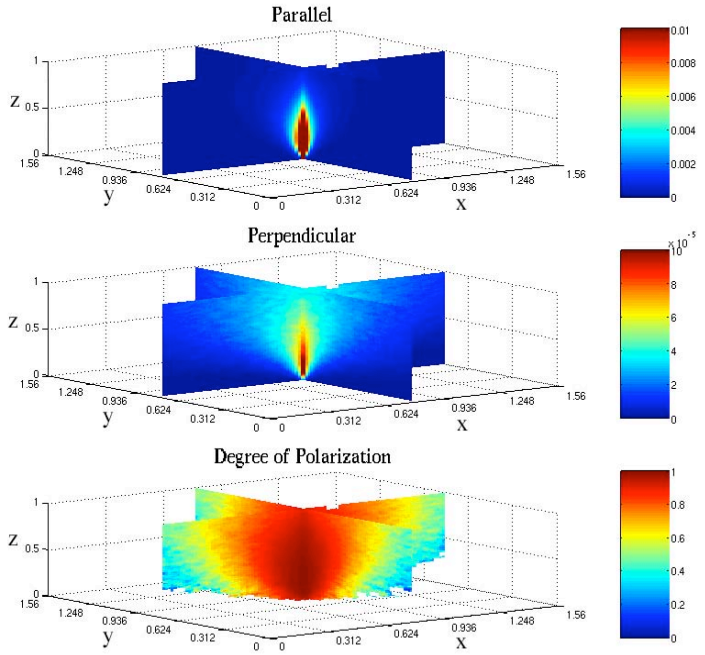

Fig. 4.20. – Transmission through various thickness z, showing x-y variation from 2000nm. μ_s=11 cm^{-1}.

Figure 4.21 and 4.22 show one plane (yz) of the previous three Figures to facilitate comparison of different sphere sizes. The 2µm spheres have strongly forward directed plots while the 0.01µm spheres have a smaller glow-ball effect for the parallel polarization that is soon lost in the background noise.

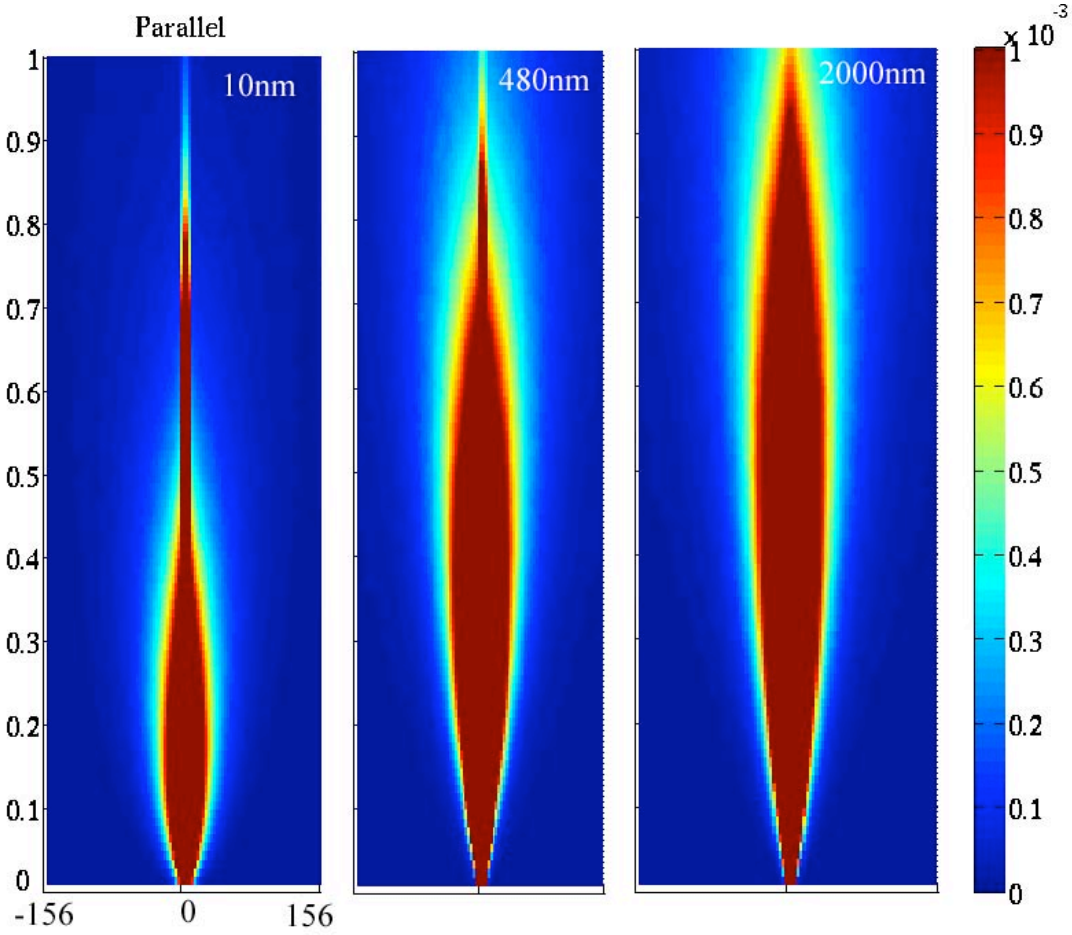

Fig. 4.21. – Co-polarized transmission for three sphere sizes.

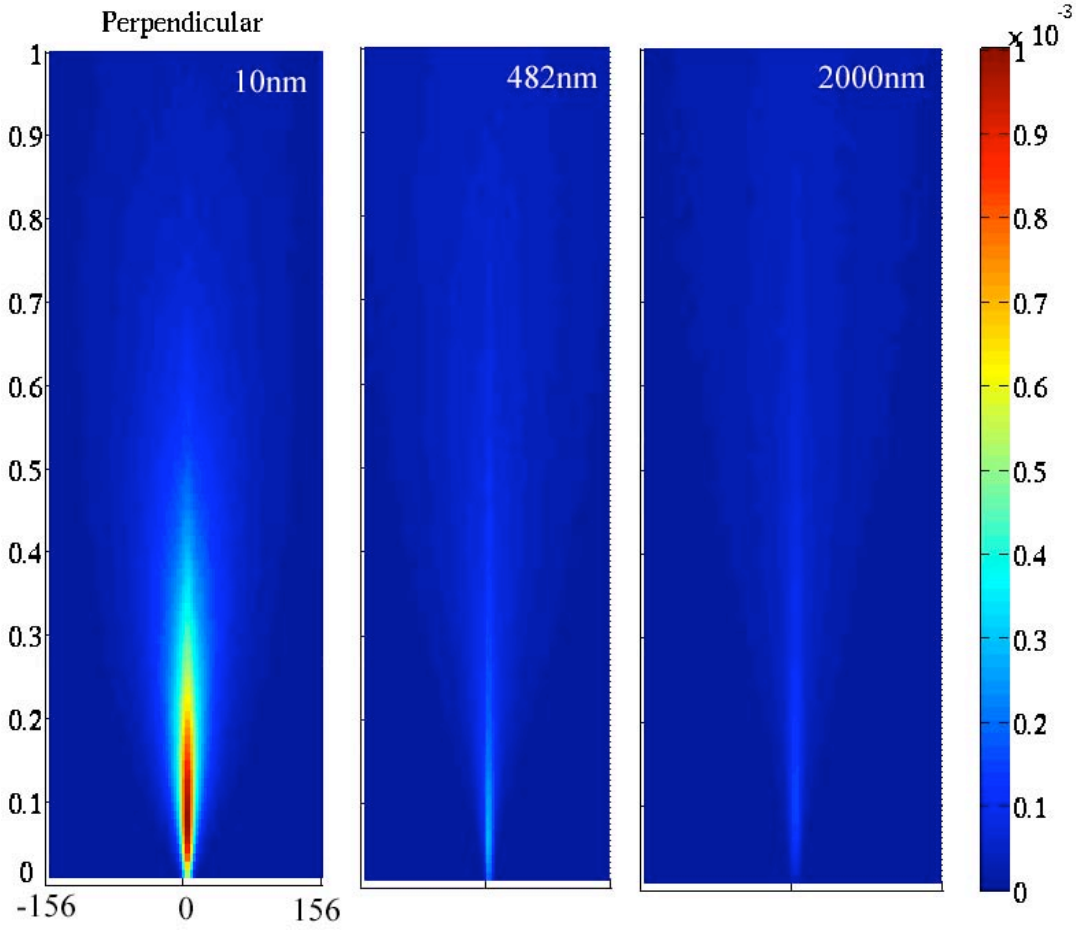

Fig. 4.22. –Cross-polarized transmission for three sphere sizes.

Figures 4.23, 4.24, and 4.25 are iso-polarization plots that give a three-dimensional representation of the degree of polarization. This type of plot, illustrate differences in the 3 dimensional structures of the degree of polarization. For the 10nm spheres a larger spread across the xz is visible when compared to yz. For images 4.23 and 4.24 all the points of the iso-polarization are equal to 0.9. The larger spheres do not have the same structure as found in Figure 4.25.

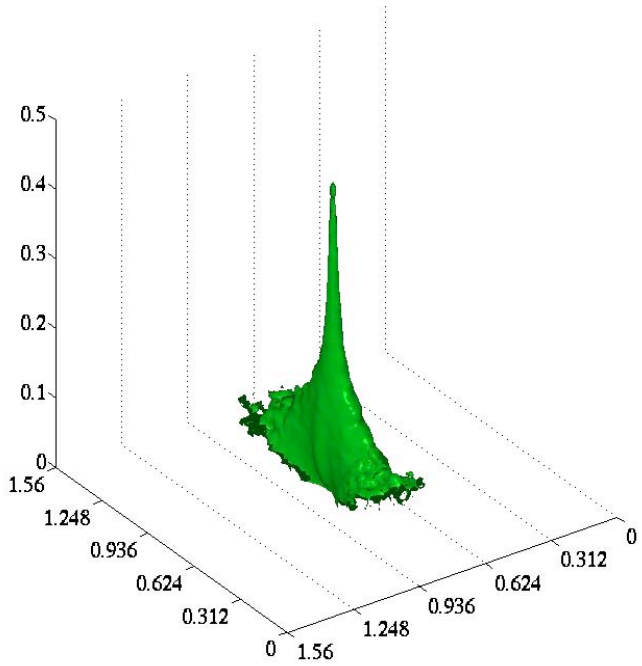

Fig. 4.23. – Iso-polarization when POL = .5 for 10nm spheres

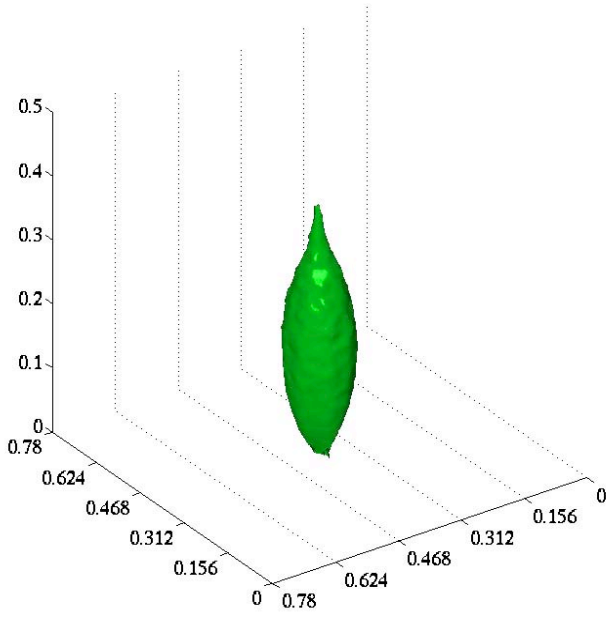

Fig. 4.24. – Iso-polarization for POL = 0.9 for 482nm spheres

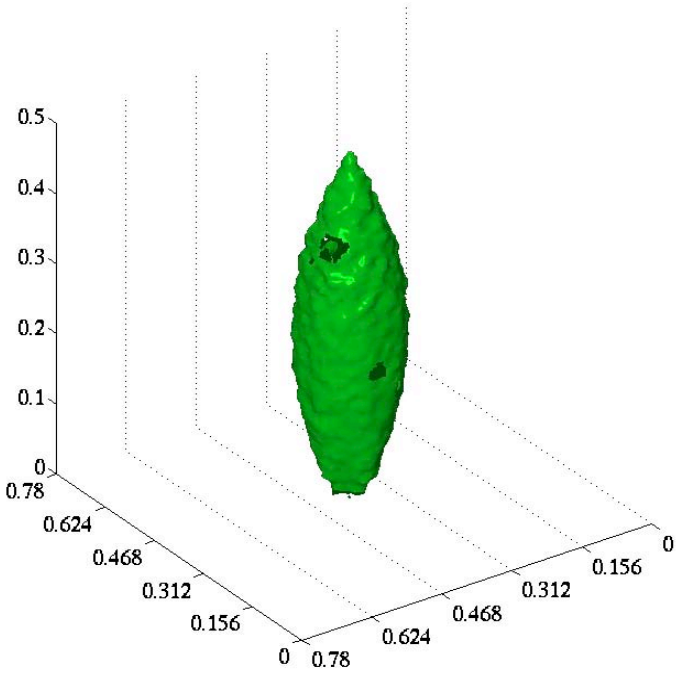

Fig. 4.25. – Iso-polarization for POL = 0.9 for 200nm spheres

In Figure 4.26 three iso-polarization surface are shown for 482nm spheres.

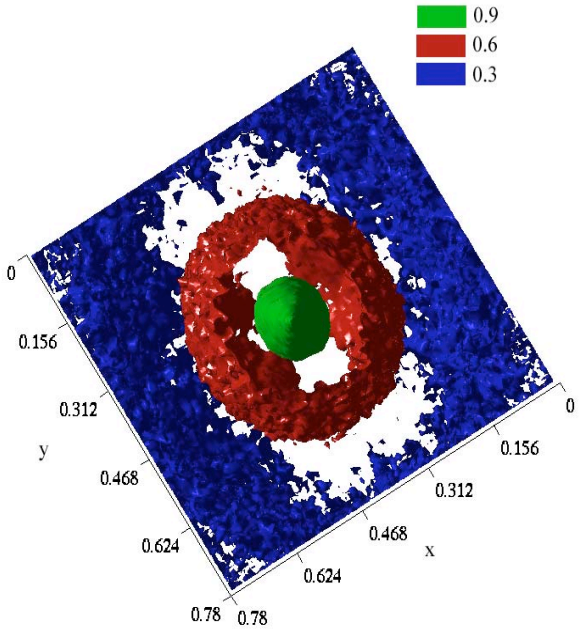

Fig. 4.26. – Three Iso-polarizations for 0.482µm spheres. Three iso-polarizations are shown: in green 0.9 in red 0.6 in blue 0.3. On the top view is visible how the scattering is not azimuthally independent. A slight cross (Maltese cross) is visible

4.5 Influence of size parameter and g in polarized light transmission through a slab

In 1994 Bicout *et al.* [93] published a paper that linked the efficiency of depolarization of solutions of microspheres to sphere size. They observed that spheres with large size parameter $x = 2\pi r/(\lambda/n \text{ medium})$ depolarize light slower that spheres with a smaller size parameter. They showed experimental and Monte Carlo models of several sphere sizes. Brosseau [19] proposed a model of depolarization of polarized light going through a slab based on an increase in entropy caused by scattering events. Brosseau also noticed that depolarization depended also on the size parameter. Simulations in this chapter also show that the size parameter affects the depolarization; with the exception

that the anisotropy is a dominant factor at when for $\tau < 30$. In Figure 4.28 Monte Carlo simulations for 5 different sphere sizes are displayed. The geometry of the experiment is the same as in Figure 4.1. A beam of light of wavelength 543nm is incident on a slab of microspheres of increasing optical thickness.

The Monte Carlo simulations kept the optical thickness ($t=\mu_s L$) constant changing the physical length (L) of the cuvette to account for different values of μ_s for different sphere sizes. The beam was polarized parallel to the optical table. Figure 4.27 show transmission for light polarized parallel to the incident beam and perpendicular to the incident beam. The parallel transmission decreases exponentially; transmission for smaller spheres decreases faster than larger spheres as expected until we compare the results for 2000nm spheres and 1069nm spheres. For these two sizes the general law doesn't seem to apply. This effect is also apparent in 4.28, and the close-up in Figure 4.29; the degree of polarization for 2000nm spheres decreases faster than the degree of polarization for 1069nm and 806nm.

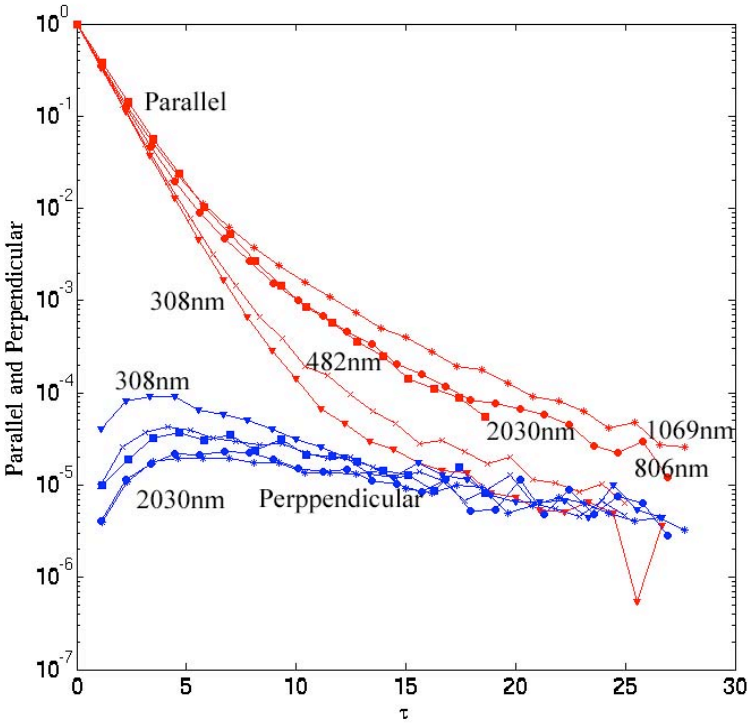

Fig. 4.27 – Parallel and perpendicular polarized transmission for slabs of increasing optical thickness. The sphere sizes considered are: 0.308, 0.482, 0.806, 1.069, and 2.03μm. The parallel polarized light is in red, the perpendicular polarized light is in blue. Notice that for the largest sphere size 2030nm, the parallel polarization decreases faster than for smaller spheres such as 1069. The anisotropy for 2030nm spheres is 0.8752 for 1069nm spheres is 0.9278. λ=543nm $n_{spheres}$ =1.59 and n_{media}=1.33.

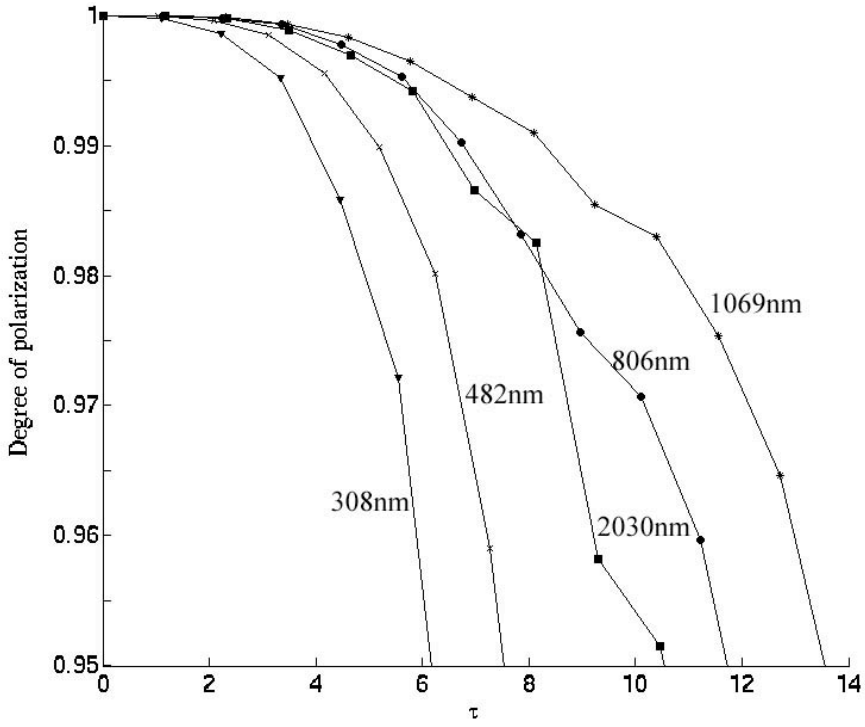

Fig. 4.28. – Degree of polarization for a beam of linearly polarized light passing through slabs of optical thickness τ. The degree of polarization for the largest sphere size 2030nm decreases faster than the degree of polarization for smaller sphere such as 1069.

The anisotropy variation as a function of sphere size, is visible in Figure 4.29. The anisotropy for 2030 nm (g=0.8752) is smaller than the anisotropy for 1069 nm (g=0.9278) and 806 nm (g=0.91532). Schmitt had studied the dependence on anisotropy for depolarization in his investigation on short-path photons in multiply scattering media [16]. His main interest was the photons spatial distribution for imaging purposes. The simulations of this work are closer to the ones of Bicout and Brosseau.

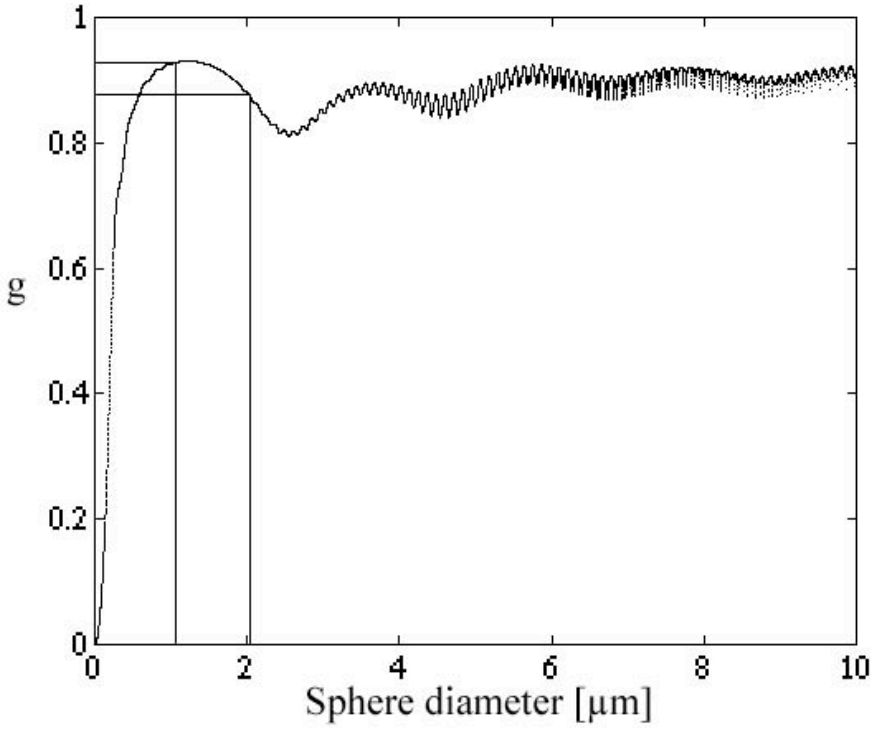

Fig. 4.29. – Anisotropy versus spheres diameter for spheres ranging from 0 to 10 μm in diameter. As pointed out in the image the anisotropy for 2030nm (g=0.8752) is smaller than the anisotropy for 1069nm (g=0.9278) and 806nm (g=0.91532).

It is important to include the anisotropy parameter g in our model. Fitting results for the numerical results of Figure 4.28 with the original heuristic model are shown in Figure 4.30, a correlation between χ and g is visible, the depolarization does not progress exponentially. For example values of χ for 2000nm spheres are higher than values of χ for 1μm spheres solutions since g for 2000nm sphere is lower than g for 1069m spheres.

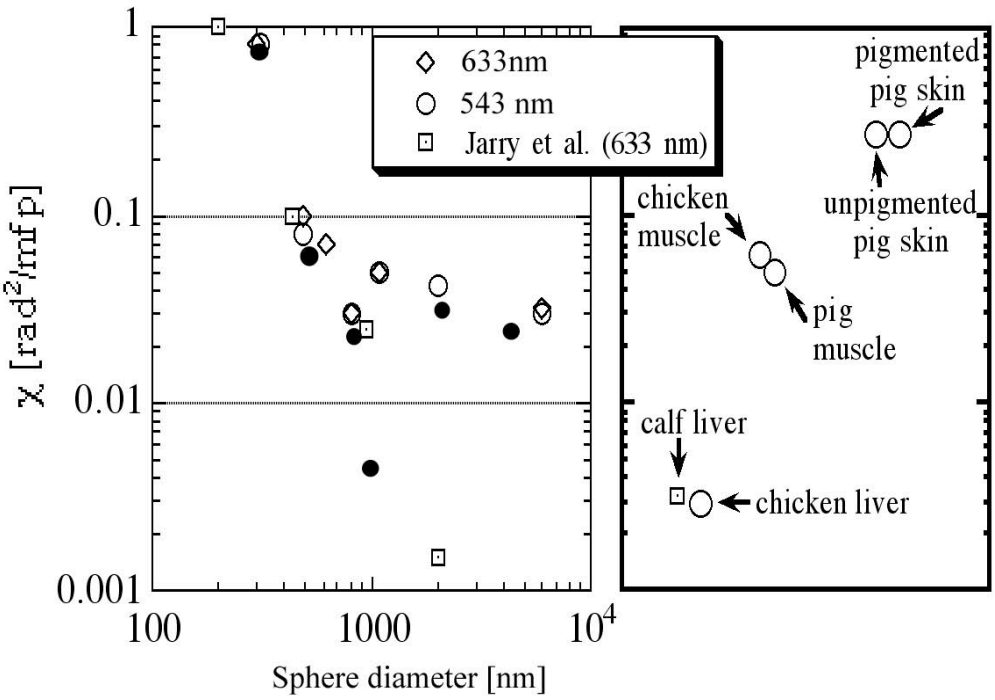

Fig. 4.30. – χ results for fitting of graphs of Figure 4.28, the value χ is strongly related to the g value

4.7 A new heuristic model of polarized light transmission through scattering media

In section 4.4.2 it was shown that the heuristic model requires a factor μ_{att} to work with apertures of different sizes. The heuristic model worked only for collimated geometry, and it was unable to model total transmission. Moreover, in 4.5 two different mechanisms for depolarization are shown, one that makes an incident polarized light beam go to a different linear status of polarization and another where the incident linearly polarized light is transformed into elliptically polarized light. It is also shown in Figure 4.13 and 4.14 how these transitions are accompanied by a change in the azimuth angle ϕ and the ellipticity angle ε. ϕ and ε have both a Gaussian distribution that broadens as the number of scattering events increase. This paragraph formulates a new heuristic model for polarized light transmission through a slab where the effect of the anisotropy of the solution is considered. The heuristic model is modified to consider the

diffused component. Comparisons with Monte Carlo simulations are performed. Four different microspheres sizes are studied: 10 nm, 300 nm 1000 nm and 2000 nm. The solutions are non-absorbing with scattering coefficient μ_s = 11 cm^{-1} and the anisotropy g dependent on the particle size. The heuristic model is built with the solution of the diffusion equation for non-absorbing scattering slabs by Prahl of equation 4.13.

$$T_{total} = \frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} = \frac{5 - \exp(-\tau + \tau g)}{4 + 3\tau - 3\tau g} \tag{4.13}$$

Where μ'_s is the reduced scattering coefficient $\mu'_s = \mu_s(1-g)$. The effect of the de-phasing of the polarized beam is added to the diffused light, similar to the original heuristic model, equation 4.9. For total transmission (i.e. without an aperture) the best approximation to the diffuse transmission is to subtract $\exp(-\mu'_s L)$ from T_{total} above. The effect of the de-phasing is added as in the previous model by multiplying the diffusion transmission by

$$\frac{I_{par\ \ scattered}}{I_o} = [\frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} - \exp(-\mu_s' L)\ ]\ \frac{1}{2}[1 + \exp(-\chi\mu_s L)] \tag{4.14}$$
$$\frac{I_{per\ \ scattered}}{I_o} = [\frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} - \exp(-\mu_s' L)\ ]\ \frac{1}{2}[1 - \exp(-\chi\mu_s L)]$$

Finally the collimated component is added to the parallel component.

$$T_{par} = \frac{I_{par}}{I_o} = \exp(-\mu_s' L) + [\frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} - \exp(-\mu_s' L)\ ]\ \frac{1}{2}[1 + \exp(-\chi\mu_s L)] + \beta \tag{4.15}$$
$$T_{per} = \frac{I_{per}}{I_o} = [\frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} - \exp(-\mu_s' L)\ ]\ \frac{1}{2}[1 - \exp(-\chi\mu_s L)] + \beta$$

Where β has been included in the model to account for imperfections in the polarizers used. The results of the new heuristic model for 10 nm microspheres are shown in Figure

4.31. The red circles are the Monte Carlo transmission (the intensity I of the total Stokes vector) for 10^5 photons passing through slabs of thickness τ, no aperture is considered in this test. The red line is the diffusion equation for a slab geometry (eq 4.12). The green circles are the parallel results for the Monte Carlo simulations where parallel =I+Q. The green line is the heuristic model of equation 4.15 where χ =0.4. Finally the blue circles are the perpendicular Monte Carlo transmission and the corresponding blue line is the heuristic model of equation 4.15.

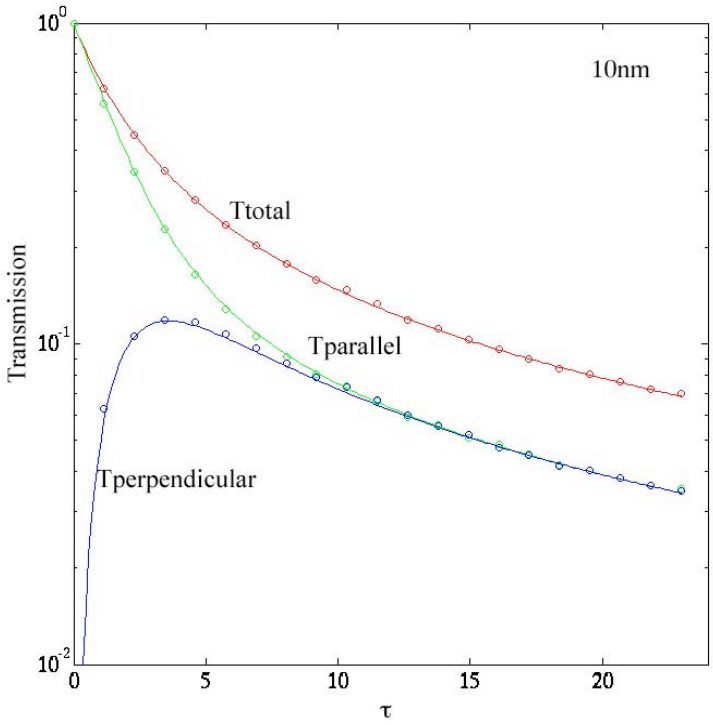

Fig. 4.31 - Monte Carlo transmission (circles) and equation 4.15 (lines) for solutions of 10nm microspheres.

Figures 4.32 and 4.33 displays the results of a Monte Carlo run for 5 sizes of microspheres. In Figure 4.33, the fits of the heuristic model to the Monte Carlo simulations of polarized transmission are shown. The value of χ from these fits is used in Figure 4.32 to show behavior of the degree of polarization. The variable b is set to zero since no imperfect polarizers are present in the Monte Carlo simulation. The errorbars were generated with a predicted error of $1/\sqrt{N}$ where $N = 10^5$.

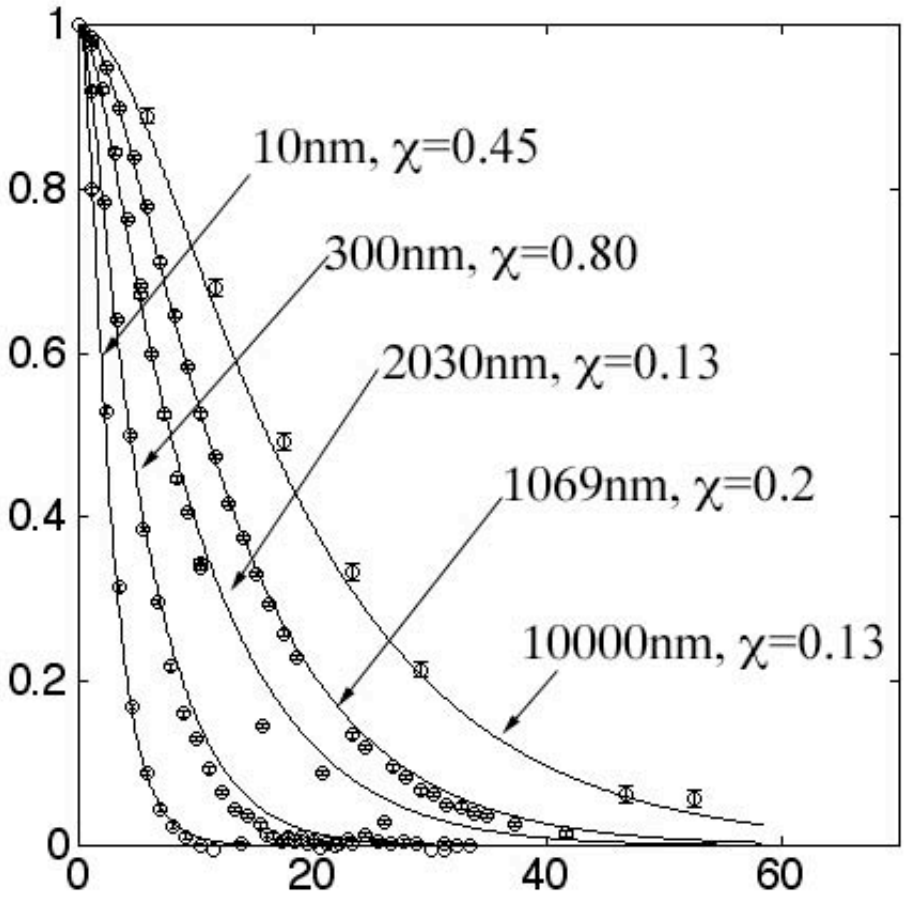

Fig. 4.32 - New heuristic model of polarized light transfer through scattering slabs. Monte Carlo runs were conducted for 4 sphere sizes (0.01μm, 0.3μm, 1.0μm, and 2.0μm). χ=0.45 for 0.01 and χ=0.8 for 0.3 μm spheres, χ=0.15 for 1μm, χ =0.2 for 2μm, and, χ =0.13.

The fact that χ for very small spheres (10nm) is smaller that χ for the larger sphere (308nm) could be attributed to the different mechanisms of depolarization (Figure 4.17) and that small spheres convert a linear state of polarization into another; while larger spheres convert linear polarization into linear and elliptical states.

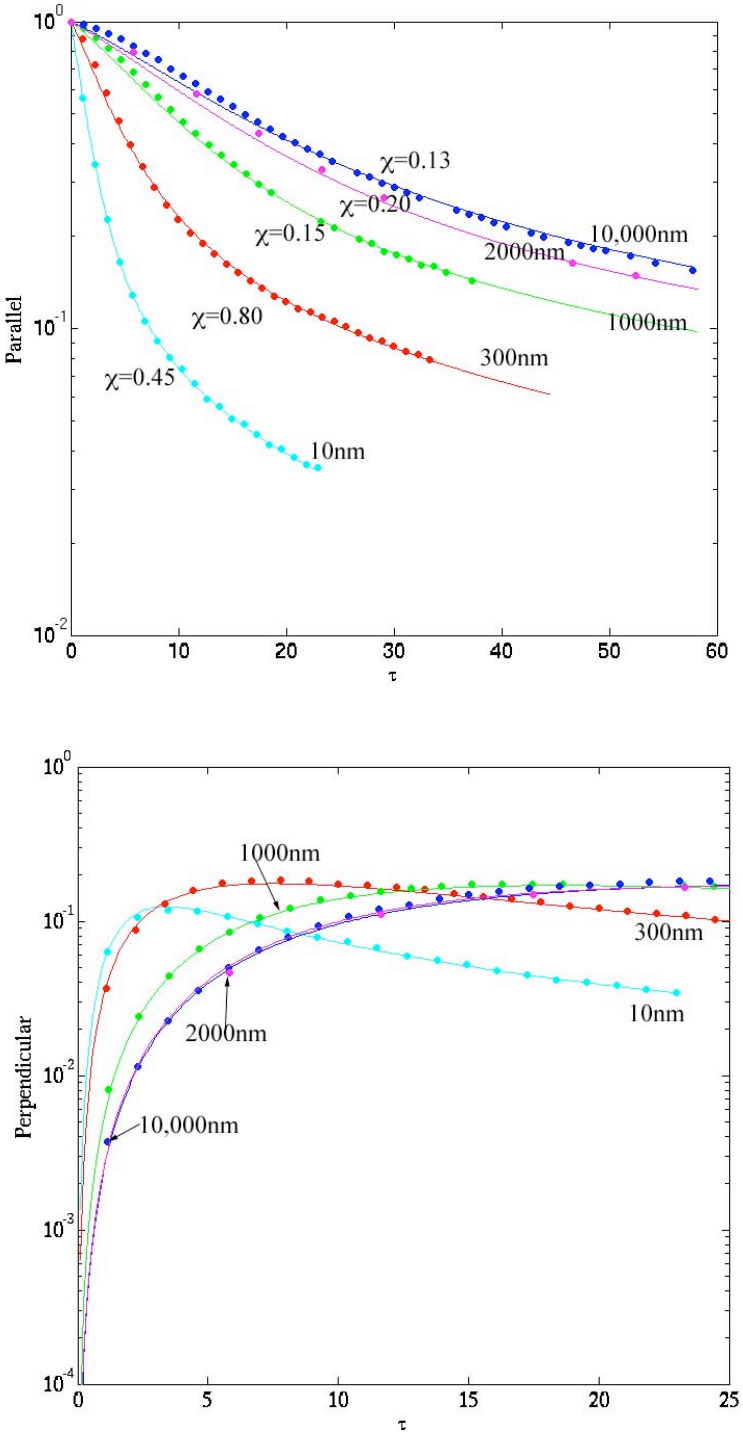

Fig. 4.33 - New heuristic model of polarized light transfer through scattering slabs. Monte Carlo runs were conducted for 5 sphere sizes Top Figure is the parallel polarized transmission the bottom Figure is the perpendicular polarized transmission

4.8 Application of the new heuristic model to a slab geometry.

The new heuristic model is modified to consider an aperture of radius a Figure 4.1. The diffused component of the beam is multiplied by a parameter *f* that depends on the radius of the aperture (a) and its distance (d) from the slab.

$$I_{par} = \exp(-\mu_s' L) + f[\frac{5-\exp(-\mu_s' L)}{4+3\mu_s' L} - \exp(-\mu_s' L)\ ]\ \frac{1}{2}(1+\exp(-\chi\mu_s L)) \tag{4.16}$$

$$I_{per} = f[\frac{5-\exp(-\mu_s' L)}{4+3\mu_s' L} - \exp(-\mu_s' L)\ ]\ \frac{1}{2}(1-\exp(-\chi\mu_s L))$$

The f parameter, was modeled as in Figure 4.34.

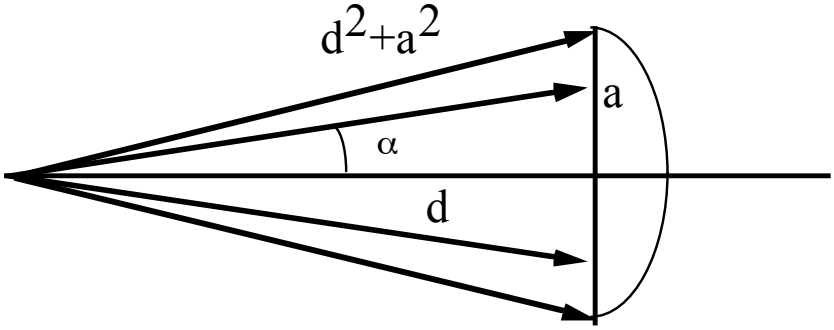

Fig. 4.34 - An aperture a is at a distance d from a slab.

$$\frac{\int_0^{2\pi}\int_0^{\alpha} T(\alpha)\cos\alpha\sin\alpha\ d\alpha\ d\phi}{\int_0^{2\pi}\int_0^{\pi/2} T(\alpha)\cos\alpha\sin\alpha\ d\alpha\ d\phi} = \frac{\det ected}{Total} \tag{4.17}$$

Let T(a)=To

$$\frac{2\pi To\int_0^{\alpha}\cos\alpha\ d\cos\alpha}{2\pi To\int_0^{\pi/2}\cos\alpha\ d\cos\alpha} = \sin^2\alpha = \frac{a^2}{d^2+a^2} \tag{4.17}$$

$$a \to \infty \quad f \to 1$$
$$a \to 0 \quad f \to \frac{a^2}{d^2} \tag{4.18}$$

The results for this aperture model are visible as dashed lines in Figure 4.35.

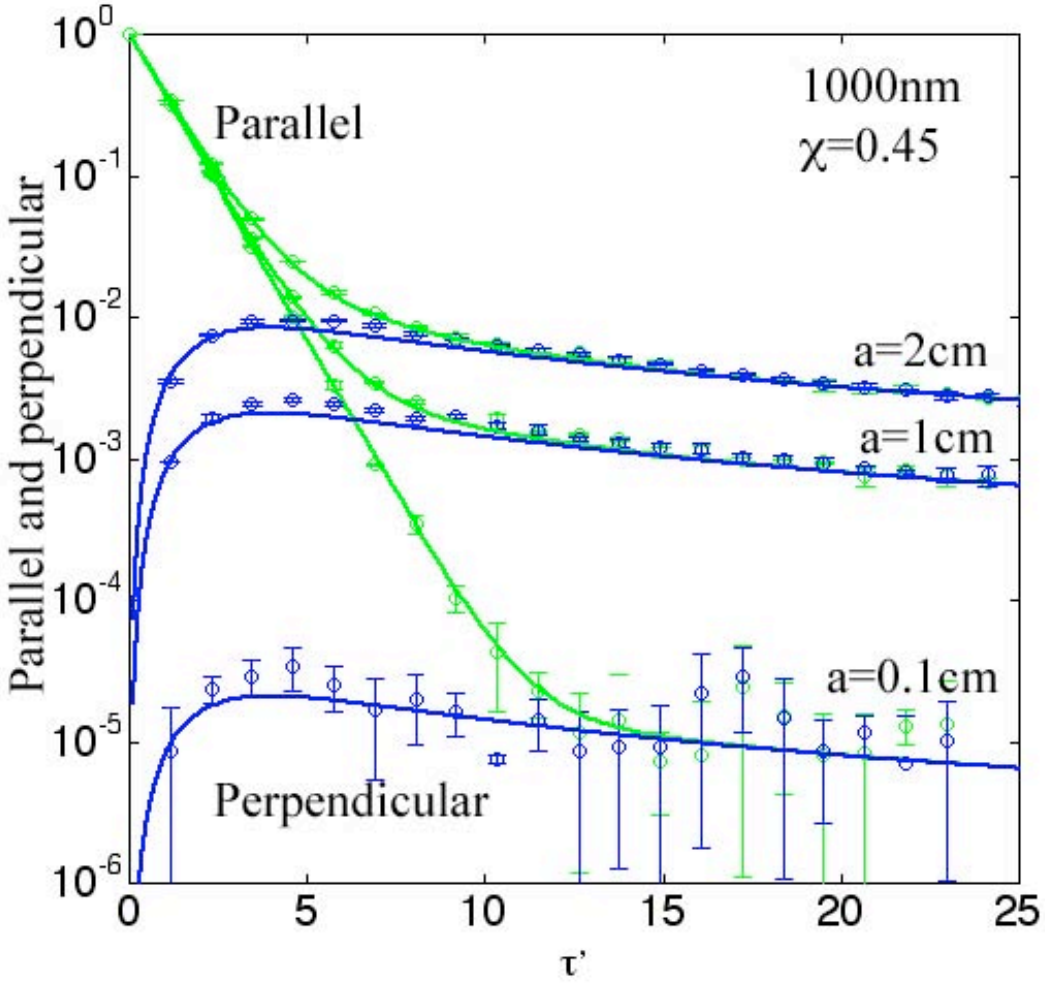

Fig. 4.35 – Monte Carlo simulations and heuristic model for the slab geometry of Figure 4.8. χ is kept constant at 0.45. Which shows that c is sensitive to sphere size but not the solid angle of collection of the detector.

This test was performed for an aperture of radius = 0.1cm for 8 different sphere sizes. The data was modeled with the new heuristic model. 10^5 photons were run for

every sphere size. For spheres larger than 0.01μm the new heuristic model had to be modified. Since light bouncing off larger sphere is forward directed more light stays collimated. Moreover the small aperture blocks most of the diffused light. The heuristic model was modified to fit these data as in equation 4.19, the parameter $\exp(-\mu_s' L)$ is replaced with $\exp(-\mu_s L)$.

$$I_{par} = \exp(-\mu_s L) + f[\frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} - \exp(-\mu_s L)\ ]\ \frac{1}{2}[1 + \exp(-\chi\mu_s L)] \tag{4.19}$$

$$I_{per} = f[\frac{5 - \exp(-\mu_s' L)}{4 + 3\mu_s' L} - \exp(-\mu_s L)\ ]\ \frac{1}{2}[1 - \exp(-\chi\mu_s L)]$$

Both models yielded similar χ values shown in Figure 4.36.

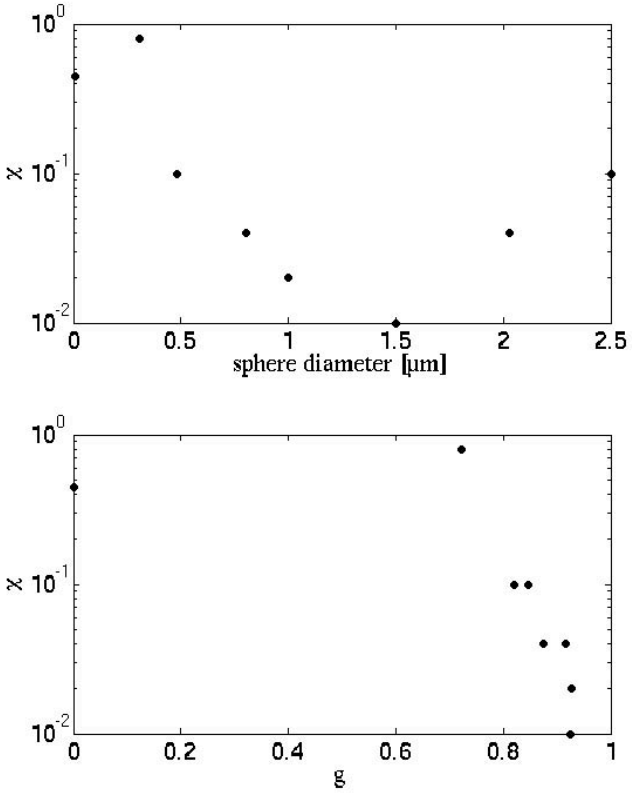

Fig. 4.36 – Results of the heuristic model fit for 8 particle sizes. In the top Figure the values of χ related to the particle size is plotted, in the bottom image the results for χ related to the anisotropy parameter g is reported

The χ factor for spheres in the Mie regime decreases as the anisotropy increases. Other groups [7] have shown experimentally that the particle size is the dominant parameter in depolarization by microspheres. These observation were made for $\tau >> 20$.

4.9 Conclusions

Two heuristic models of polarized light transfer through a slab of microspheres have been described in this chapter. The first model is based on two parameters χ and μ_{att} works well for biological media and solutions of microspheres. This model relies on the collimated component of the incident beam, the experiments and Monte Carlo simulations were performed with a slab geometry and an aperture in front of a detector to limit the amount of diffused light collected. Its main drawback was the need for two fitting parameters χ and μ_{att}. The second model based on the diffusion equation performs very well when the modeled experimental geometry has no apertures and total transmittance is collected. To model experiments where the transmitted light is mostly collimated the model has to be modified as in equation 4.16. Both model yielded similar results for the χ parameter. For $0< \tau < 30$ the χ parameter decreases as the anisotropy increases. In this chapter I also show how the mechanism of depolarization for small and large scatterers are different. Rayleigh (small) particles redirect an incident linearly polarized light beam to different states of linear polarization while Mie (large) scatterers can transform a linear incident state to an elliptical state of polarization.

The anisotropy parameter g is an important parameter in the depolarization of a photon passing through solutions of microspheres. As observed in chapter 3 with Monte Carlo simulations, large spheres will depolarize faster than smaller sphere if their anisotropy is smaller. For example a 1μm spheres solution (g = 0.9278 @ 543 nm) depolarizes light slower than 2.03 μm (g = 0.8752 @543 nm). A preliminary curve of g versus χ Figure 4.36 shows that the depolarization changes rapidly for g>0.7. I think that the trend in between 0 and 0.7 will be constant corresponding. This could be due to the isotropic scattering and the fact that only balistic (with unchanged polarization) reach the detector. Further studies will be necessary to clarify this point.

Chapter 5
Experimental analysis of polarized light travel into biological tissues

5.1 Introduction

Polarized light filtering has been used to reduce glare, as in the case of polarized sunglasses and certain dermatologic tools, and to enhance contrast when looking through scattering media, such as haze, clouds or fog. Applications to enhance contrast range from underwater imaging [94] to medical imaging [27, 28]. Walker *et al.* [95] proposed a method based on cross polarization imaging to improve the visibility of absorbing objects buried in scattering media. Chang *et al*. [96] used the same technique and demonstrated that contrast and visibility depth could be improved if the scattering media was composed of Rayleigh particles, but less improvement was obtained in the case of Mie particles. For biological tissues, Studinski and Vitkin [88] studied polarized light in the exact backscattered direction, showing that the degree of polarization in that direction, and in that direction only, plateaus at a value larger than zero.

Polarized light has been used in many different applications, from imaging of skin cancer to stress and strain measurements on biological tissues; an overview of the many applications is given in chapter 1. Of particular interest is how deep polarized light will

travel before completely losing its original polarization. A first set of experiments studied this problem. In this experiment a roughened metal band was embedded into deeper and deeper slabs of tissues, and back-reflected light was collected with a camera until the mirror completely disappeared in the tissue. The behavior of polarized light was analyzed for several visible wavelengths.

A second set of experiments used Mueller matrix formalism to determine some biological tissue parameters such as retardance and birefringence. This set of experiments was conducted in transmission. Three biological tissues were studied; calf liver, chicken muscle, and porcine skin.

In the following paragraphs a numerical value for the depth of depolarization of polarized light is developed. Measurements in biological tissues are difficult to make and effected by the environment conditions such as temperature and humidity. A definitive difference among different tissues types (skin, muscle, liver) was often observed. Some experimental values of depth of penetration are given.

5.2 Reflection of a rough metal band embedded into a scattering media

5.2.1 Materials and Methods

The experimental set-up is shown in fig 5.1. Thin tissue samples were prepared by slicing frozen biological samples (porcine skin, calf liver, chicken muscle) with a razor blade; the thickness was denoted as L. All samples were obtained at a local grocery store. A 16-bit digital camera (Princeton Scientific, TEA/CCD-1024 TKM/1) was positioned normal to the sample. A white light source (Omron, Schaumburg, IL) was positioned at an angle of ~ 20 degrees to the sample/camera axes; the emitted beam was polarized with a polarizer (Ealing, Electro-Optics, Holliston, MA) with nominal extinction 10^{-6}. A single wavelength (± 10nm) was selected with the use of a tunable filter (Varispec^{TM}, CRI, Woburn, MA). The tunable filter had also an on-board polarizer, polarized parallel to the optical table. The orientation of the light source was chosen to avoid the specular glare from the air/glass/tissue boundary.

A rough metal band was obtained from a thin film of aluminum whose surface was coarsed with sandpaper. Each thin sample was sequentially placed on top of the metal band laying on the background tissue slab.

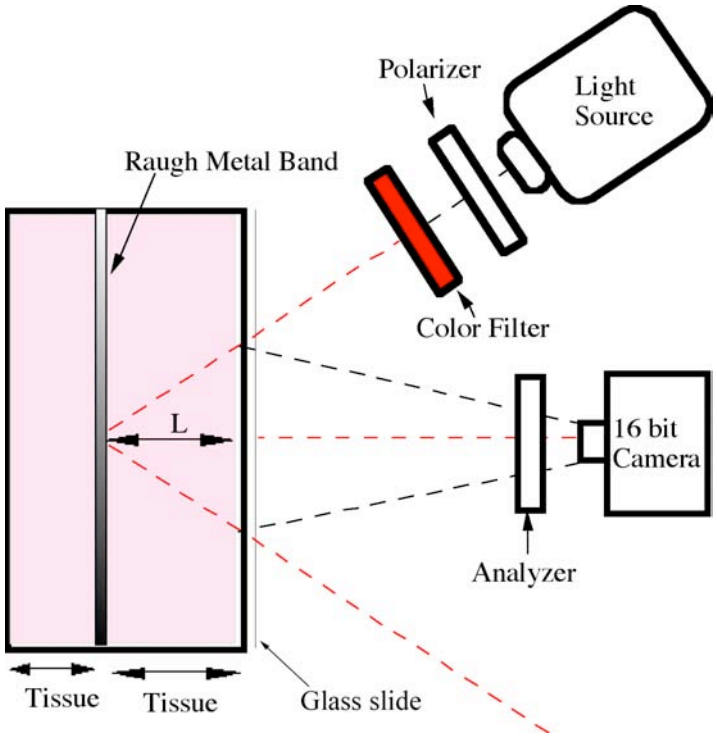

Fig. 5.1. - Experimental apparatus. Illumination light was linearly polarized parallel to the scattering plane that corresponds to the source/tissue/camera triangle. The analyzer was a linearly polarizer oriented first parallel then perpendicular to the scattering plane yielding two images, denoted PAR and PER, respectively. A rough metal band was placed on a thick tissue slab and an overlaying thin tissue section of thickness L was placed over the metal band.

A glass slide was placed over the sample and was optically coupled to the tissue by a drop of saline. For every tissue thickness two images were taken with the detector analyzer oriented either parallel (PAR) or perpendicular (PER) to the source polarization. The images were manipulated algebraically using MATLAB^{TM} software to yield the PAR – PER image. Regions of the PAR – PER images were chosen manually that lay either on top of the metal band or on top of a region of tissue without a metal band, denoted by subscripts $_{metal}$ and $_{tissue}$. The mean value and standard deviation of the pixel values in the regions of interest were determined, and the following difference value was calculated:

$$D = (PAR - PER)_{metal} - (PAR - PER)_{tissue} \tag{5.1}$$

This difference would initially be a high number for a very thin overlaying tissue (small L). However, as L increased, value of D would approach zero as $(PAR - PER)_{metal}$ would become equal to $(PAR - PER)_{tissue}$ because the metal band would be buried so deeply that polarized light reflected by the metal would be randomized.

The values D versus L were plotted and fit with an exponential curve:

$$D = Ae^{-L/z_e} \tag{5.2}$$

Initially, the values of A and z_e were fit by least squares fitting for each of the 3 tissues. Then the mean values of A at each wavelength from these 3 fittings were chosen and the data D refit using a fixed value of A for each wavelength experiment. This protocol was followed because A should not vary from tissue to tissue but should be a function of the wavelength dependences of the light source power, the metal reflectivity and the camera sensitivity. The mean values of A were 32000, 63700, 25700 and 11200 for the wavelengths 480, 543, 633 and 696 nm, which were high compared to the values of D from the images. Consequently, the fitted values for z_e were relatively insensitive to any variations in A.

To estimate the error in the estimated values of z_e, the choice of z_e was varied over a broad range (0 – 600 μm) and used in Eq. 5.2 to yield a predicted value of D called pD. An error score was calculate based on the sum of the squared differences:

$$error = \sum_{i=1}^{N_L} (pD_i - D_i)^2 \tag{5.3}$$

and a probability score was estimated based on the inverse of this error score:

$$p(z_e) = \frac{1}{error} \tag{5.4}$$

When properly normalized, this $p(z_e)$ constitutes an estimate for the probability density function in units of [μm^{-1}]. The best estimate of z_e equaled the value of z_e for which $p(z_e)$ was maximum. The full-width half-max (FWHM) of the $p(z_e)$ for each wavelength was determined. For this report, the p(ze) was normalized such that its peak value was equal to 1 to facilitate graphic comparisons.

5.2.2 Results

Figure 5.2 presents some of the raw images

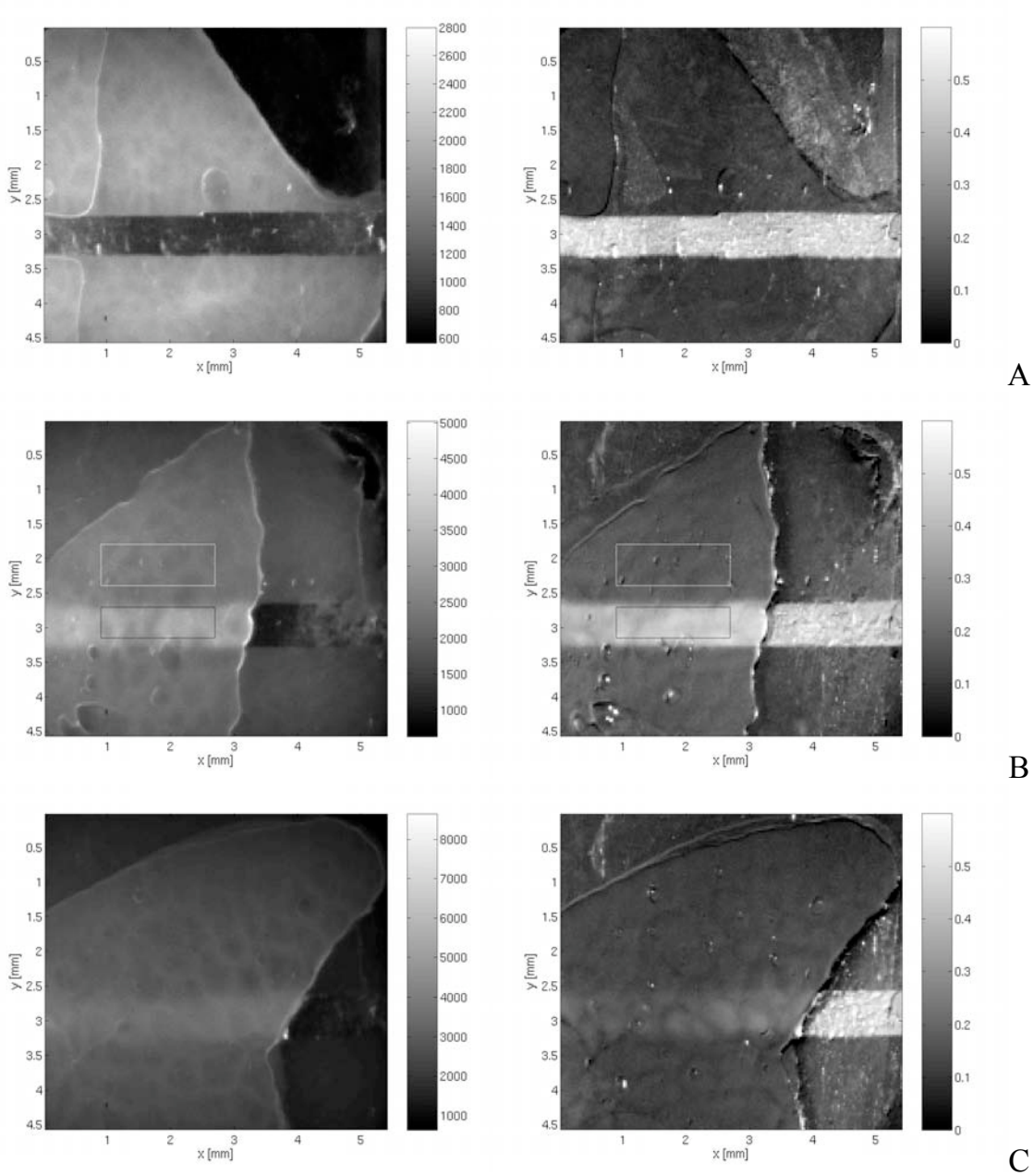

Fig. 5.2. - Images of Pol = (PAR-PER)/(PAR+PER). Liver at 696 nm wavelength. Left column shows PER images, and right column shows POL images. (A) Metal band with no overlaying tissue (L = 0). (B) Metal band covered on left by overlaying tissue of thickness L = 420 μm. Also shown are two rectangular regions of interest, one over metal and one not over metal. (C) Metal band covered on left by overlaying tissue of thickness L = 1020 μm.

Figure 5.3 shows the raw values PAR_{metal}, PAR_{tissue}, PER_{metal} and PER_{tissue} as a function of the thickness L of the overlaying tissue layer.

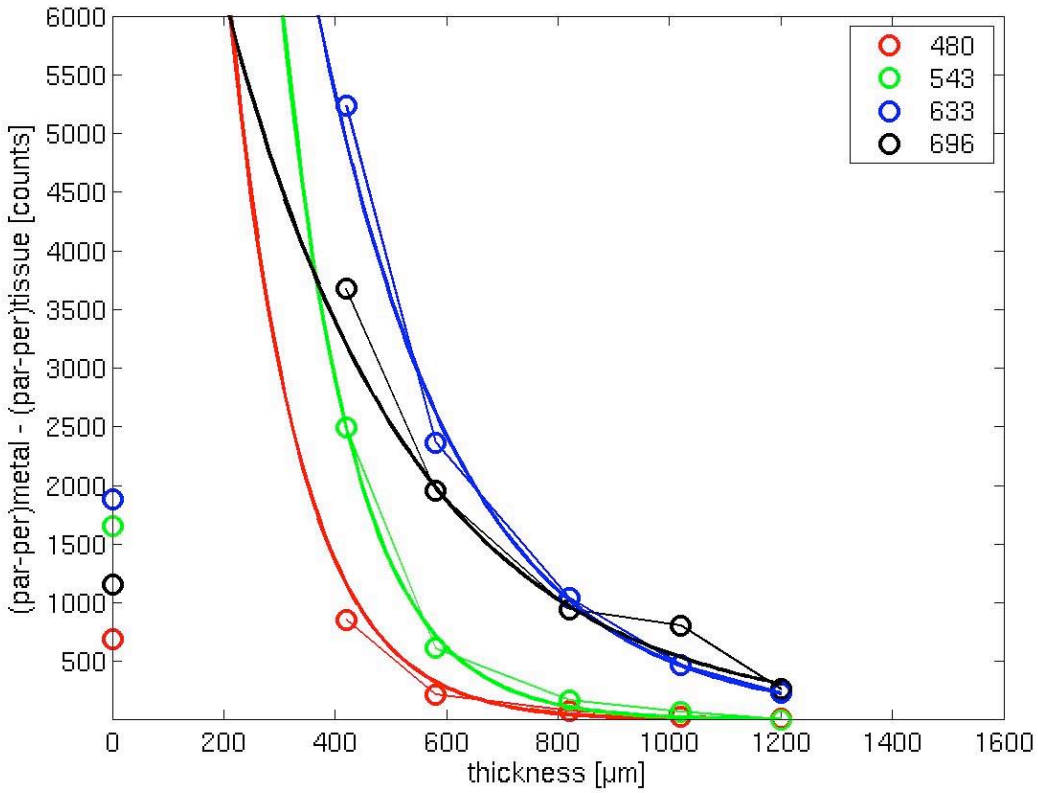

Fig. 5.3. - Example of experimental data. The tissue was liver. The values of D = (PAR-PER)metal – (PAR-PER)tissue versus thickness L of the tissue that overlays the metal band. Data for four wavelengths (480, 543, 633, and 696 nm) shown. Data at zero thickness is low because incident light is obliquely reflected directly off metal and misses the camera.

Figure 5.4 shows the values of $D = (PAR - PER)_{metal} - (PAR - PER)_{tissue}$ as a function of L. Also shown as solid lines are the curve fits based on equation 5.2 that specify z_e.

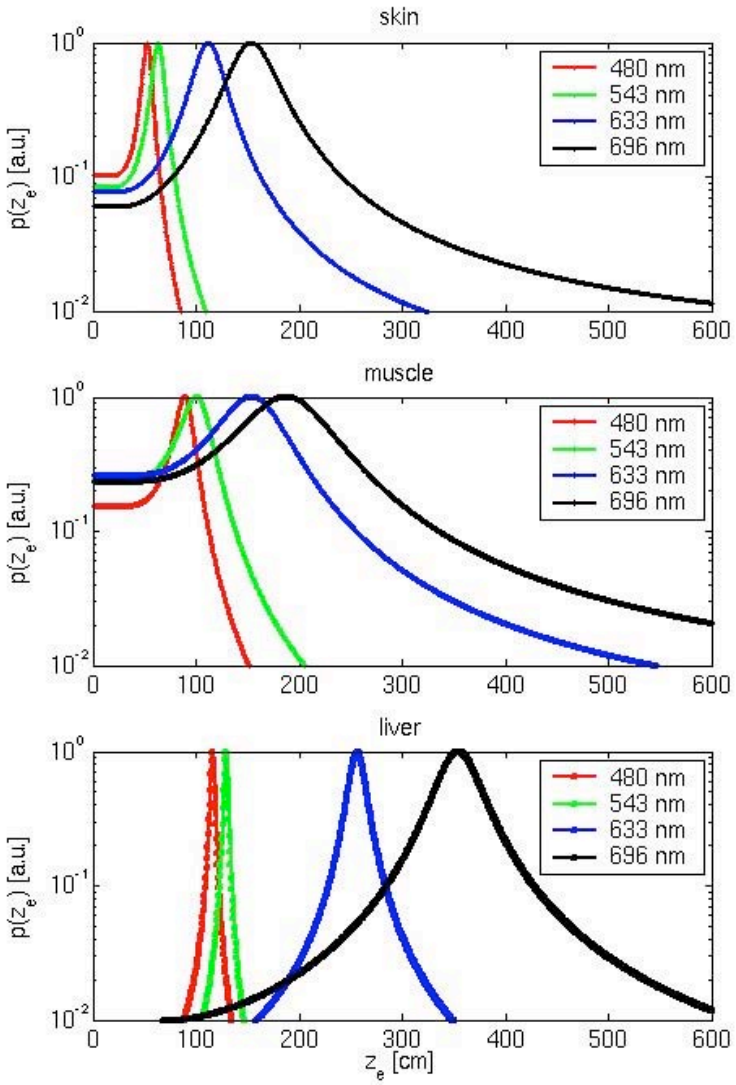

Fig. 5.4. - Probability density functions for estimate of 1/e depth of imaging, $p(z_e)$. The choice of z_e was varied from 0 to 600 μm and the prediction of $D = (par - per)_{metal} - (par - per)_{tissue}$ was calculated. A function $p(z_e)$ was calculated equal to 1/error where error equaled the sum of squared differences between experimental data and predicted values for D. The $p(z_e)$ was normalized to 1 at its peak value to aid comparisons, consequently the units of $p(z_e)$ are in arbitrary units. The three graphs show the skin, muscle and liver curves at each of the four wavelengths tested.

Figure 5.5 presents these estimated values of $2z_e$ along with error bars, equal to ±FWHM, whose total length depicts the full-width half-max of the estimate of $2z_e$. The depth of imaging increased with increasing wavelength and in the order of skin < muscle < liver. The estimate $2z_e$ equaled 106, 178 and 232 μm at 480 nm and 307, 374 and 709 μm at

696 nm for skin, muscle and liver, respectively. The error of the estimate $2z_e$ was about ±6% of the estimated value of $2z_e$ based on the FWHM of Fig. 5.3.

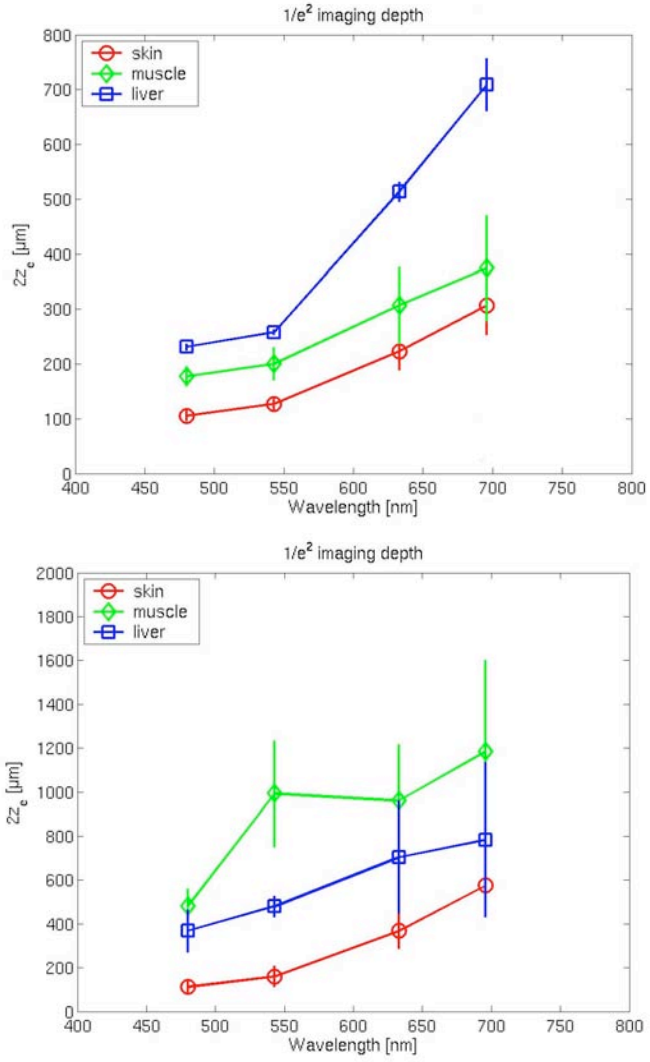

Fig. 5.5. - The $1/e^2$ depth of imaging (= $2z_e$) as a function of wavelength for three tissues (skin, muscle, liver). The values are the values of 2ze corresponding to the peaks of Fig. 5.4. The error bars indicate the full-width half-max of the distribution for the estimate of $2z_e$.

Figure 5.5 shows wavelength dependence of the depth of imaging for all three tissue samples, highly birefringent tissue depolarizing light faster (skin>muscle>liver) than less birefringent ones.

A beam of polarized light depolarizes very quickly in the considered tissue samples, shallow depth of penetration is used to focus the images in the very superficial layers of the skin. One could argue that using short wavelengths the same effect could be achieved. Although the depth of imaging is similar for polarized and unpolarized imaging in the 400-500 nm range the contrast in the polarized images is much higher. Figures 5.6 and 5.7 illustrate this point. In Figure 5.6 the left column corresponds to images of 420 μm liver samples, the left image is of 1200 μm liver samples. Three types of images are studied, simple PER images (A), PAR-PER (B), and POL(C). A simple visual inspection shows that the PAR-PER images and POL image have higher contrast that the simple PER image. To obtain a more quantitative assessment of the achieved image contrast, regions of interest in the background tissue are compared with region if interest for on the metal band for the PER and PAR-PER images (red squares on Figure 5.6). In Figure 5.7 the results of this test are shown. The contrast for the PAR-PER image is twice as large as the contrast for the PER image.

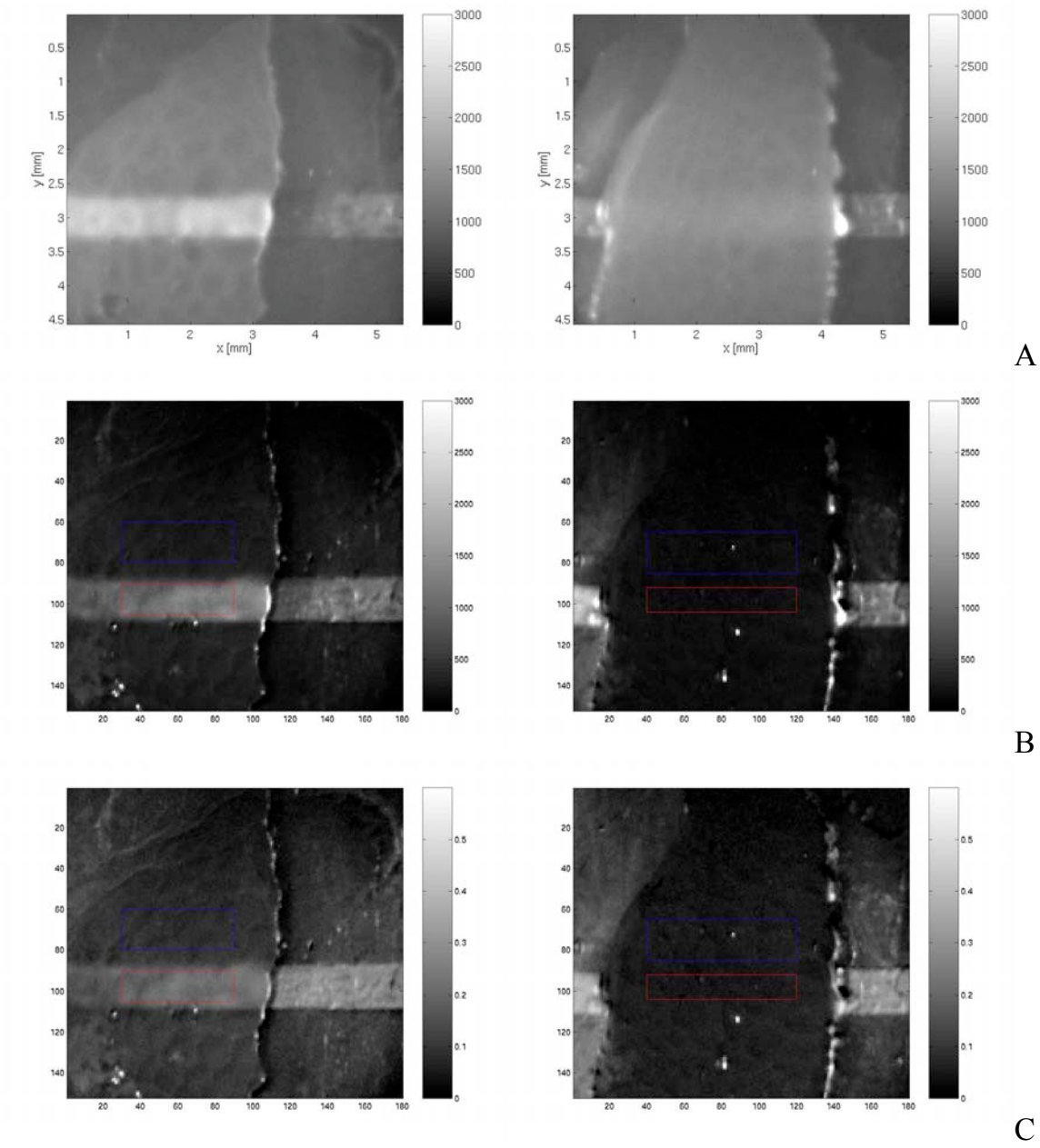

Fig. 5.6. - Left column is L = 400µm. Right colume L = 1200µm. (A)PER. (B) PAR-PER. (C) POL

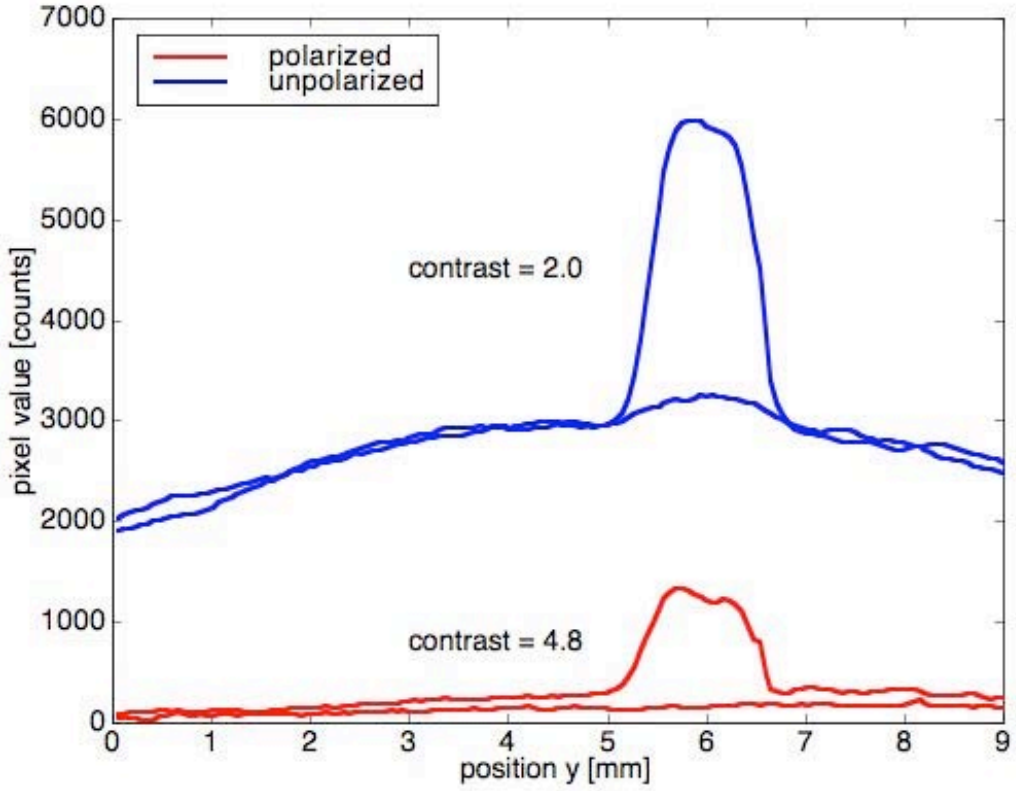

Fig.5.7. – Contrast for polarized and unpolarized images. The blue curve obtained comparing areas of background tissue and area on the rough metal band for PER images, the red curve was obtained comparing the same areas in PAR-PER images. The contrast for PAR-PER images is more that twice as big as the PER images contrast. Polarized light imaging aids in situations where the depth of imaging is shallow and enhances contrast in the superficial layers of the skin.

5.3 Mueller Matrix description of collimated light transmission through liver, muscle, and skin

Propagation of polarized light through liver, muscle, and skin was studied using the Mueller Matrix formalism. Collimated He-Ne laser light was passed through a set of polarization elements to create one of four possible polarization states; linear horizontal (H), linear vertical (V), linear 45-degree (P), and right circularly (R) polarized light. The beam passed through thin sections of tissue of varying thickness (0.2-0.9 mm thick). The unscattered, collimated, transmitted light passed through a second set of polarization elements to analyze of each of the 4 possible polarization states (H,V,P,R). The transmitted intensities for the 16 possible combinations of source and detector polarization yielded a data matrix that was converted into a Mueller matrix describing the propagation properties of the tissue. The results could be modeled as if the tissue were ideal retarders whose birefringent values

$$\Delta n = \frac{\Delta\lambda}{2\pi L} \tag{5.5}$$

where L is the thickness of the sample and Δn were in the range of 0.0001 to 0.0005 which is consistent with the literature. The order of the strength of birefringence was liver < muscle < skin. Although the above birefringence values may apply to muscle, the structure of liver and skin are not necessarily consistent with the ideal retarder model.

5.3.1 Introduction

Mueller calculus was first introduced in 1945 as a simplified method of solving problems involving polarizers and retarders [18]. Mueller calculus employs matrices and vectors. The matrices are called Mueller matrices and specify optical components. The vectors are called Stokes vectors and define the status of polarization of the incident and transmitted beam of light. Multiplying the incident beam vector by the Mueller matrix of

a sample yields the status of polarization of the transmitted beam [48, 98]. If different tissues are modeled by their Mueller matrices then the status of polarization of light traveling through the tissues can be predicted.

The on-axis unscattered collimated transmission of chicken liver, muscle, and skin was measured. The results were converted into two parameters as a function of sample thickness: (1) the degree of linear polarization, and (2) the degree of circular polarization. These experimentally derived values were compared with theoretically predicted curves of the degrees of linear and circular polarization. The results were roughly consistent with tissue behaving as an idealized retarder whose birefringence was comparable to values from the literature.

5.3.2 Materials and methods

The experimental apparatus is shown in Figure 5.8. A HeNe laser (Melles-Griot) emitting wavelength 632.8 nm was chopped and passed through a polarizer (Ealing) oriented parallel to the optical table. The polarized beam was passed through a half-wave plate and a quarter-wave plate (Meadowlark Optics) whose fast axes were manually controlled to generate sources with four different polarizing states, 0˚, 45˚, 90˚ and right circularly polarized. A sample of known thickness was located normal to the incident beam. A quarter-wave plate and a polarizer in front of a silicon detector were sequentially rotated to collect a the transmitted light with a specific status of polarization: 0˚, 45˚, 90˚ and right circularly polarized. The detector was 1cm in diameter and was positioned at 90 cm from the sample to collect primarily the unscattered collimated component of the transmitted light. The data were collected with a lock-in amplifier (Stanford Research SR830) via a GPIB interface to a computer (G4, Macintosh).

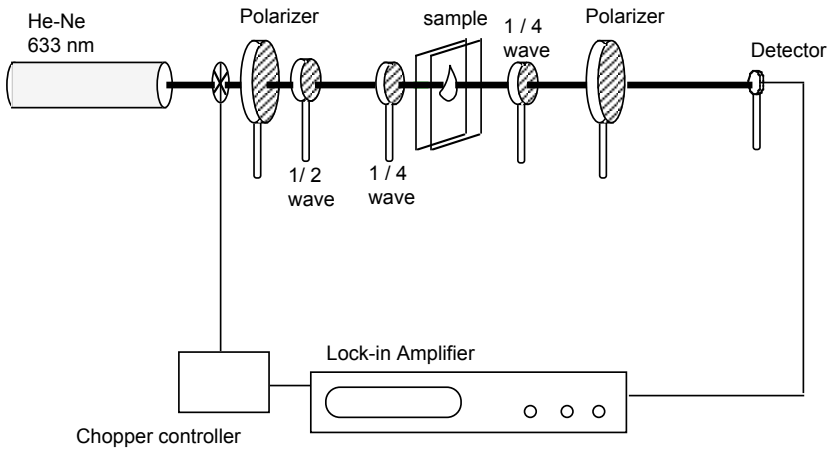

Fig. 5.8. - Experimental apparatus

The experimental apparatus was calibrated with samples having known Mueller Matrices such as: polarizers, half-waveplates and quarter-wave plates. Measurements of transmitted intensity were conducted on tissues samples of known thickness. Tissues (calf liver, chicken muscle and porcine skin) were frozen then cut in the desired thickness and sandwiched between glass slides. The status of polarization of linearly or circularly polarized light transmitted through a tissue sample was measured for each of the 16 combinations of source and detector polarization status. The Stokes Vector for linearly polarized light was multiplied by the experimentally specified Mueller Matrices. The resulting transmitted Stokes vector, [I, Q, U ,V], was used to calculate the degree of linear polarization,

$$P_L = \frac{\sqrt{(Q^2 + V^2)}}{I} \tag{5.6}$$

and the degree of circular polarization,

$$P_C = \frac{\sqrt{V^2}}{I} \tag{5.7}$$

5.3.3 Tissue preparation

Three different light-scattering tissues, calf liver, pig skin, chicken muscle, were measured. Fresh tissues samples from the abattoir were frozen then cut to the desired thickness (0.2-0.9 mm) and placed between glass slides. Sample thicknesses were measured by caliper while held between the glass slides, accounting for the slide thicknesses. The muscle fibers were oriented 45 degrees relative to the surface of the optical table. The skin was oriented with the laser beam normal to the skin surface such that the collagen fibers were perpendicular to the beam but the azimuth angle of fibers was randomly distributed around the beam axis normal to the fibers. Liver had no preferred orientation. Measurements of transmitted intensity were conducted on tissue samples of the various known thicknesses. Five locations on every sample were measured, giving five distinct Mueller matrices per sample. Three samples per tissue type were considered. Five thickness were considered for skin and size for muscle and liver.

5.3.4 Modeling of Mueller matrix polarimeter

All optical components of the experimental set-up were modeled with the corresponding Mueller matrices derived from separate experiments [92]. To generate each of the four needed detectors, the Mueller Matrices for an appropriately oriented polarizer and quarter-wave plate were used. In the same way for each of the four needed sources, the Mueller Matrices of an appropriately oriented quarter-wave plate, half-wave plate and linear polarizer were used. The train of elements used in the experiment is shown in Figure 5.12. LP stands for linear polarizer and QW for quarter-wave plate.

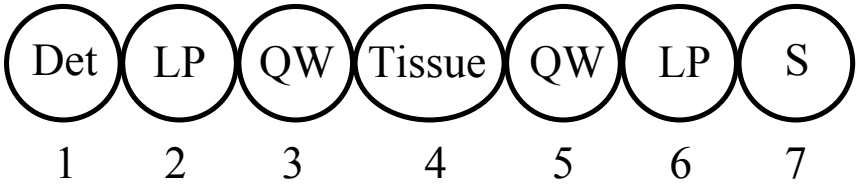

Fig. 5.9. – Train of polarizing elements used for Mueller matrix type experiments. LP stands for linear polarizer QW for quarter-wave plate. The orientation of the polarizing elements is described in table 5.1

Alternatively a train of matrices such as in Figure 5.10 can be used where the orientations of the source polarizers (0˚,45˚ and 90˚) are achieved using an half-wave plate (element 6). The drawback of this set up is the introduction of a new polarizing element and the consequent experimental error. The advantage is that the source polarization is maintained constant with the polarizer (element 7) so avoiding any artifact due to the bias in polarization of the light source (element 8).

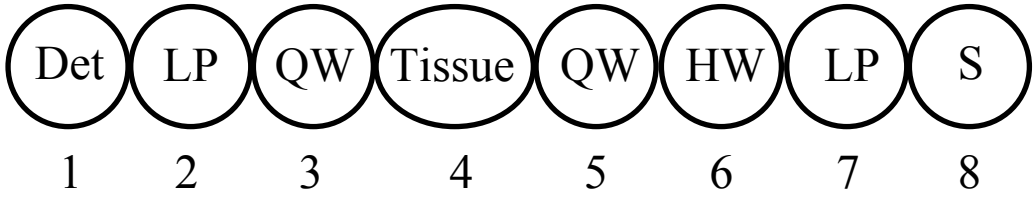

Fig. 5.10. – Alternative train of polarizing elements used for Mueller matrix type experiments. LP stands for linear polarizer QW for quarter-wave plate and HW for half-wave plate. The orientation of the polarizing elements is described in table 5.1

The orientation of the polarizing elements for the set up descibed in Figure 5.9 is shown in table 5.1.

	1	**2**	**3**	**4**	**5**	**6**	**7**
HH	D	$\alpha = 0^\circ$	-	S	-	$\alpha = 0^\circ$	S
HV	D	$\alpha = 90^\circ$	-	S	-	$\alpha = 0^\circ$	S
HP	D	$\alpha = 45^\circ$	-	S	-	$\alpha = 0^\circ$	S
HR	D	$\alpha = 0^\circ$	$\phi = 45^\circ$	S	-	$\alpha = 0^\circ$	S
VH	D	$\alpha = 0^\circ$	-	S	-	$\alpha = 90^\circ$	S
VV	D	$\alpha = 90^\circ$	-	S	-	$\alpha = 90^\circ$	S
VP	D	$\alpha = 45^\circ$	-	S	-	$\alpha = 90^\circ$	S
VR	D	$\alpha = 0^\circ$	$\phi = 45^\circ$	S	-	$\alpha = 90^\circ$	S
PH	D	$\alpha = 0^\circ$	-	S	-	$\alpha = 45^\circ$	S
PP	D	$\alpha = 90^\circ$	-	S	-	$\alpha = 45^\circ$	S
PV	D	$\alpha = 45^\circ$	-	S	-	$\alpha = 45^\circ$	S
PR	D	$\alpha = 0^\circ$	$\phi = 45^\circ$	S	-	$\alpha = 45^\circ$	S
RH	D	$\alpha = 0^\circ$	-	S	ϕ=45	$\alpha = 0^\circ$	S
RV	D	$\alpha = 90^\circ$	-	S	ϕ=45	$\alpha = 0^\circ$	S
RP	D	$\alpha = 45^\circ$	-	S	ϕ=45	$\alpha = 0^\circ$	S
RR	D	$\alpha = 0^\circ$	$\phi = 45^\circ$	S	ϕ=45	$\alpha = 0^\circ$	S

Table 5.1 – Table of polarizing element positions to construct Mueller matrices.

The Mueller matrices for a linear polarizer oriented at an angle α and a quarte-wave plate oriented at an angle ϕ are shown in equation 5.8. S is the Stokes vector for the incident beam. The incient beam is unpolarized.

$$LP = \frac{1}{2}\begin{bmatrix} 1 & \cos 2\alpha & \sin 2\alpha & 0 \\ \cos 2\alpha & \cos^2 2\alpha & \sin 2\alpha \cos 2\alpha & 0 \\ \sin 2\alpha & \sin 2\alpha \cos 2\alpha & \sin^2 2\alpha & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$QW = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos^2 2\phi & \sin 2\phi \cos 2\phi & 0 \\ 0 & \sin 2\phi \cos 2\phi & \sin^2 2\phi & -\cos 2\phi \\ 0 & \sin 2\phi & \cos 2\phi & 0 \end{bmatrix} \qquad (5.8)$$

$$S = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

In equation 5.9 the train of matrices necessary to generate a horizontal source and a perpendicular detector, i.e. an HV measurement, is shown.

$$\begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} [sample] \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \end{vmatrix} \qquad (5.9)$$

The experimental measurements correspond to the intensity, I, which is the first element in the transmitted Stokes vector. The theoretical model of the apparatus consisted of sixteen different matrix multiplications like the above using different matrices to generate the needed sources and detectors. The first term (I) of every

Stokes vector, corresponding to the intensity of the exiting beam, was arranged in a matrix form as shown in Figure 5.11. This matrix was called the *Data matrix*.

	H	V	P	R
H	HH	HV	HP	HR
V	VH	VV	VP	VR
P	PH	PV	PP	PR
R	RH	RV	RP	RR

Fig. 5.11 - Data matrix (DM). Each element corresponds to the intensity of transmitted collimated light when the source was either H, or V, or P or R and the detector was either H, or V, or P or R.

From the Data matrix and using the transformation proposed by Yao and Wang [85], the Mueller matrix of the sample was obtained. The transformations are restated in the following table:

M11 =HH+HV+VH+VV
M12 = HH+HV-VH-VV
M13 = 2*PH+2*PV-M11
M14 = 2*RH+2*RV-M11
M21 = HH-HV+VH-VV
M22 = HH-HV-VH+VV
M23 = 2*PH-2*PV-M21
M24 = 2*RH-2*RV-M21
M31 = 2*HP+2*VP-M11
M32 = 2*HP-2*VP-M12
M33 = 4*PP-2*PH-2*PV-M31
M34 = 4*RP-2*RH-2*RV-M31
M41 = 2*HR+2*VR-M11
M42 = 2*HR-2*VR-M12
M43 = 4*PR-2*PH-2*PV-M41
M44 = 4*RR-2*RH-2*RV-M41

Table 5.2 – Transformations necessary to form Mueller matrix from Data matrix

$$MM = \frac{1}{2}\begin{bmatrix} M11 & M12 & M13 & M14 \\ M21 & M22 & M23 & M24 \\ M31 & M32 & M33 & M34 \\ M41 & M42 & M43 & M44 \end{bmatrix}$$

Fig. 5.12 - Reconstructed Mueller matrix, all the elements are obtained from table 5.2

The transmission value of the filter used in the experiments was varied for different tissue thicknesses and the values are listed below. A value of 1 means no filter was used. A value of 0.001 means an OD3 neutral density filter was used.

Skin

Thickness (μm)	170	240	370	
Filter transmission	1	1	0.001	

Muscle

Thickness (μm)	150	270	360	420
Filter transmission	1	1	1	1

Liver

Thickness (μm)	200	360	420	900
Filter transmission	1	1	0.001	0.001

The un-scattered beam in the absence of any tissue, yield a transmitted intensity value of 3.8 V in laboratory units. Measurements of transmitted intensity were recorded in mV, then divided by 3.8 V, then divided by the transmission value of the filter to yield tissue transmission [dimensionless]. These transmission values are listed in Appendix C.

5.3.5. Calibration and error analysis of the Mueller matrix polarimeter

Every optic component of the set up was tested experimentally to confirm manufacturing specifications. For the polarizer (Ealing Inc., Tucson, AZ) a Malus' law type experiments where conducted, moreover the real extinction of the polarizer was tested and was equal to 10^{-5}, manufacture specification were 10^{-6} extinction. For quarter-wave plates the ability to generate circularly polarized light at a specific wavelength was measured. For the utilized $\lambda/4$ retarder (Meadowlark, Frederic, CO) the retardance accuracy was $\lambda/350$ according to specification, in our measurements the retardation accuracy was confirmed retardation was 158.1 (nm) instead of the specified 159.76 nm.

Several tests using only air as a sample were conducted, Figure 5.13. The first matrix on the left is the ideal Mueller Matrix for air, which is an identity matrix because air does not have any polarizing property. Experimentally the Mueller Matrix for air

deviated from its ideal behavior as shown in the matrix in the center due to the deviation from ideality of the various optical components in the system.

$$Ideal = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Experimental = \begin{bmatrix} 0.98 & 0.01 & 0.02 & 0.01 \\ 0.01 & 0.98 & 0.02 & -0.01 \\ 0.07 & -0.04 & 0.86 & -0.08 \\ 0.02 & -0.01 & -0.04 & 0.86 \end{bmatrix}$$

Fig. 5.13 - (Left) Ideal Mueller matrix of air. (Center) Experimental Mueller matrix for air (Right) Corrected Mueller Matrix of air which accounts for the non-ideality of the optical components of the system.

The Mueller matrix for air was used to calibrate the system. Two different calibration techniques were implemented.

In the first approach a minimization algorithm was used to determined the correction matrix, MM_{cal}. For a known incident Stokes vector $[I, Q, U, V]_{incident}$, MM_{cal} was computed to obtained predetermined values of $[I, Q, U, V]_{transmitted}$

$$\begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix}_{transmitted} = MM_{\det_pol} MM_{cal} MM_{air_\exp} MM_{source_pol} \begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix}_{incident} \tag{5.10}$$

In a second approach a correction matrix was determined which allowed

$$\begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix}_{transmitted} = MM_{\det_pol} MM_{air_\exp} MM_{cal} MM_{source_pol} \begin{vmatrix} I \\ Q \\ U \\ V \end{vmatrix}_{incident} \tag{5.11}$$

to generate the Data Matrix of transmitted intensities expected for an air sample.

By placing MM_{cal} before or after MM_{air}, the errors in the apparatus were assigned either to the source or to the detector. The truth was probably somewhere in between but the goal was to identify the extreme possibilities and see how differently the two models behaved. The predicted behavior was not strongly sensitive to whether Eq. 5.10 or 5.11 was used.

The corrected Mueller matrix for an air sample was:

$$Corrected\ \ Mueller\ \ Matrix\ \ of\ \ Air = \begin{bmatrix} 0.97 & 0 & 0 & 0 \\ 0 & 0.97 & 0 & -0.014 \\ 0 & 0 & 0.97 & -0.015 \\ -0.02 & 0 & 0 & 0.92 \end{bmatrix}$$

This approach are not too different from calibration technique pointed out in the literature [98]. For example Hauge uses a Direct Method (16-Intensities), and a *Calibration Response Matrix* whose elements are combination of experimental air measurement.

Measures conducted with ellipsometers often are burdened by error in the alignment or imperfection of the various components.

The literature is rich of error analysis techniques for this kind of measurements since the correct evaluation of a Mueller Matrix on the sample is dependent on the quality of the system. [98, 99]

The error matrices [100] for every element of the polarimetric system and for element combinations can be generated. The error matrix for a complete system is:

$$\delta\mathbf{M} = \delta\mathbf{M_x} + \delta\mathbf{M_d} = \mathbf{M}\,\delta\mathbf{m_x} + \delta\mathbf{m_d}\,\mathbf{M} \qquad (5.12)$$

where $\delta\mathbf{m_x}$ and $\delta\mathbf{m_d}$ are the error matrix due to the source and detector respectively, similar to what was used in the calibration system.

In the case of air equation 5.12 reduces to

$$\delta\mathbf{M} = \delta\mathbf{m_x} + \delta\mathbf{m_d} \qquad (5.13)$$

Several cases are presented here analyzing the various part of the system.

For a system as in Figure 5.14 there are two kind of error: 1 misalignment errors and 2 depolarization errors. Since the polarizers were of high quality (Tper/Tpar = 10^{-5}) and considering a measure in air, only the misalignment error is important.

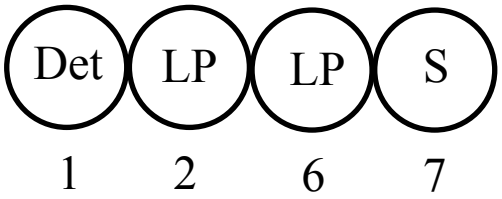

Fig. 5.14. – First example, a system is composed of a detector (D) two linear polarizer (LP) and source (S)

The error matrix for this error is given by:

$$\delta m_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2\delta P \\ 0 & 2\delta P & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\delta m_y = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\delta A \\ 0 & -2\delta A & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where δP and δA are small rotation misalignment of the polarizer and analyzer respectively. In an HH measurement the error is only in the U component of the [I, Q, U, V] vector as shown in Figure 5.15. For a range of polarization misalignment of ± 3˚ the error in Q can be quite high. In my measures the uncertainty in positioning angle was ±1˚ giving an error of less than 2%.

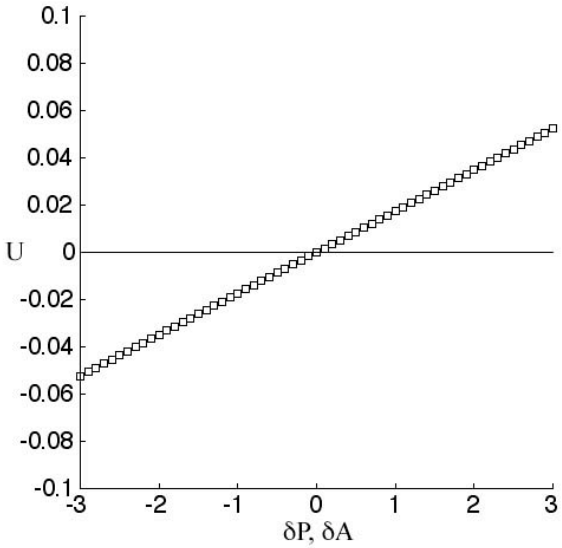

Fig. 5.15 – Uncertainty in an HH measurement, the only affected component is the U component. In the range of 1 the uncertainty is less than 2%

When retarders are added to the system the situation becomes more complex. For a quarter wave plate in the source arm with retardation $\tau = 90^\circ + \delta\tau_c$. In the source system for $C=0^\circ|90^\circ \pm C$ and $C=45^\circ \pm C$ being the orientations of the retarder.

$$\delta m_x(C = 0\,|\,90) = \begin{bmatrix} D_c + D_p & 0 & 0 & 0 \\ 0 & D_c - 2D_{vc} & -2\delta C & \mp 2(\delta C - \delta P) \\ 0 & 2\delta C & 0 & \mp\delta\tau_c \\ 0 & \mp 2(\delta C - \delta P) & \mp\delta\tau_c & 0 \end{bmatrix}$$

$$\delta m_x(C = \pm 45^\circ) = \begin{bmatrix} D_c + D_p & 0 & 0 & 0 \\ 0 & 0 & -2\delta C & \mp\delta\tau_c \\ 0 & 2\delta C & D_c - 2D_{vc} & \mp 2(\delta C - \delta P) \\ 0 & \pm\delta\tau_c & \pm(\delta C - \delta P) & 0 \end{bmatrix}$$

where Dc are the depolarization and the cross-polarizers depolarization, D_p and D_{vp} are the depolarization and cross-depolarization only for the polarizers. For a perfect retarder Dc and Dvc are 0, for a perfect polarizer Dp and Dvp are 0. The same matrices can be

built for the detector arm. Adding all the errors the element of the error matrix as shown by Song are:.

$$\delta M_{oo} = 0$$
$$\delta M_{01} = \delta M_{02} = \delta M_{03} = \delta M_{10} = \delta M_{20} = \delta M_{30} = \delta M_{33} = dM_{11} = dM_{22} = 0$$
$$\delta M_{12} = 2\delta Q - 2\delta C + 2(\delta C - \delta P)\cos t_c + 2(\delta A - \delta Q)\cos t_q$$
$$\delta M_{13} = -\delta M_{31} = -\delta t_c \sin 2C - 2(\delta C - \delta P)\cos 2C \sin t_c + \delta t_q \sin 2Q - 2(\delta A - \delta Q)\cos 2Q \sin t_q$$
$$\delta M_{21} = 2\delta C - 2\delta Q - 2(\delta C - \delta P)\cos t_c - 2(\delta A - \delta Q)\cos t_q$$
$$\delta M_{23} = -\delta M_{32} = \delta t_c \cos 2C - 2(\delta C - \delta P)\sin 2C \sin t_c - \delta t_q \cos 2Q - 2(\delta A - \delta Q)\sin 2Q \sin t_q$$

The depolarization effects are considered less important that the misalignment effects all the depolarization effect were kept in between 0 and .1 since all the optic components were of good professional quality. The main error was attributed to the rotational stages. With the error related to the polarizer was ±3 degree and the quarter wave plates positioning error was ±1 degree.

The calculate Air+error matrix becomes:

$$Air + error = \begin{bmatrix} 0.9736 & -0.0055 & -0.0022 & 0.0079 \\ 0.0033 & 0.9642 & -0.0016 & -0.0032 \\ -0.0000 & 0.0018 & 0.9635 & -0.0163 \\ -0.0137 & 0.0089 & -0.0046 & 0.9587 \end{bmatrix}$$

Minimization algorithms are often used to characterize the error of a system; in future work the minimization approach will be used to minimize error sources.

5.3.6 Results

Figure 5.16 shows the transmission versus sample thickness for the three tissue types: skin, muscle and liver.

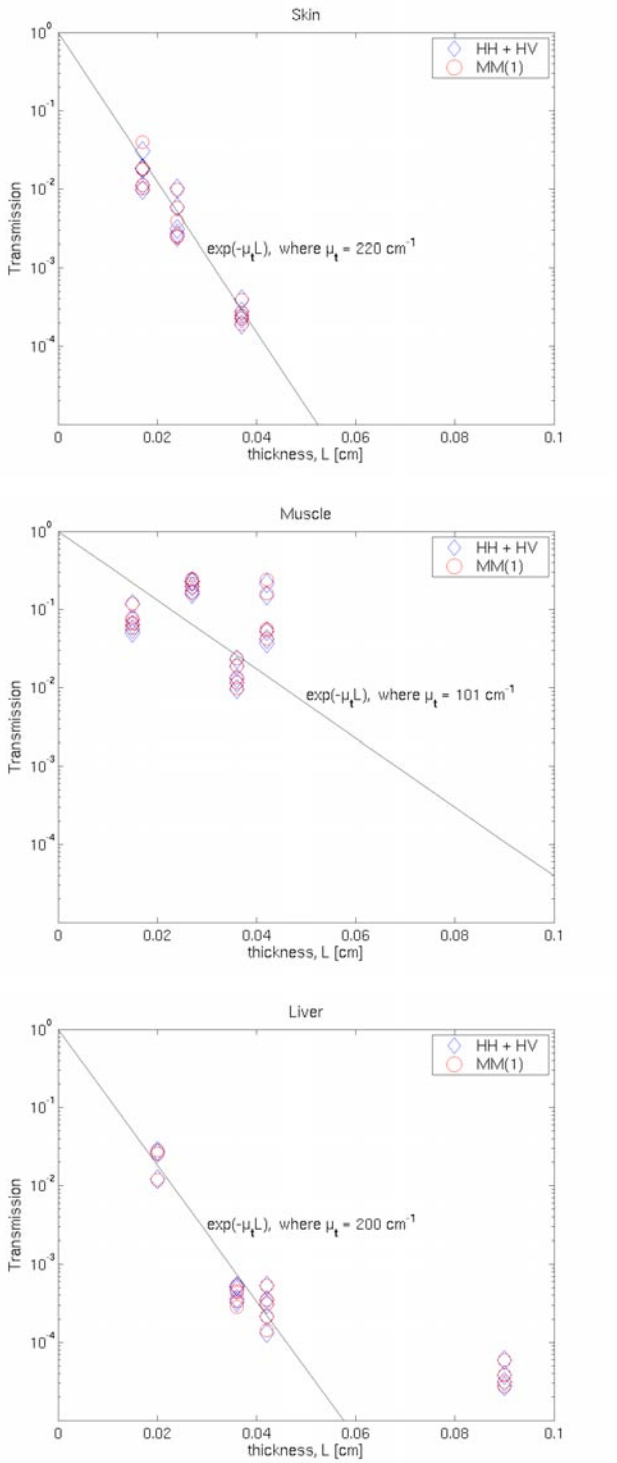

Fig. 5.16. - Transmission versus tissue thickness. (Top) Skin (porcine). (Middle) Muscle (chicken breast). (Bottom) Liver (chicken).

The transmission is calculated in two ways. First, the value of (HH + HV)/2 is plotted (blue diamonds). Second, the value of the first element of the Mueller Matrix MM(1,1) is plotted.

Figure 5.17 to 5.22 shows both the Data Matrix and the Mueller Matrix for the three tissue types.

Each element of the Data Matrix or Mueller Matrix is an x-y matrix of values, composed of the number of tissue samples of differing thickness tested (x axis) and the number of different positions measured on each sample. The data are shown as the $\log_{10}(T)$, with the color bar indicating the 6 decades of transmission values (10^{-6} to 1). Similarly, each element of the Mueller Matrix shows an x-y matrix of values denoting tissue thickness and position of measurement. For each of the sample thicknesses and sample positions measured, the matrix elements are normalized by the value of M(1,1) so that MM(1,1) = 1. The color bar indicates values of MM(i,j) ranging between –1 to +1.

SKIN

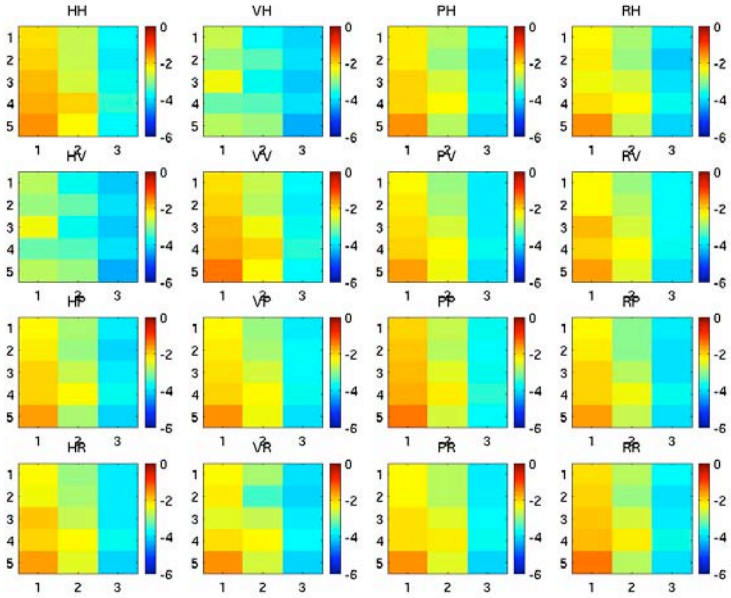

Fig. 5.17. - Data Matrix for 5 positions (y = 1 to 5) on 3 skin samples of thickness 170, 240 and 370 mm (x = 1 to 3, respectively). Color map = $\log_{10}(T)$.

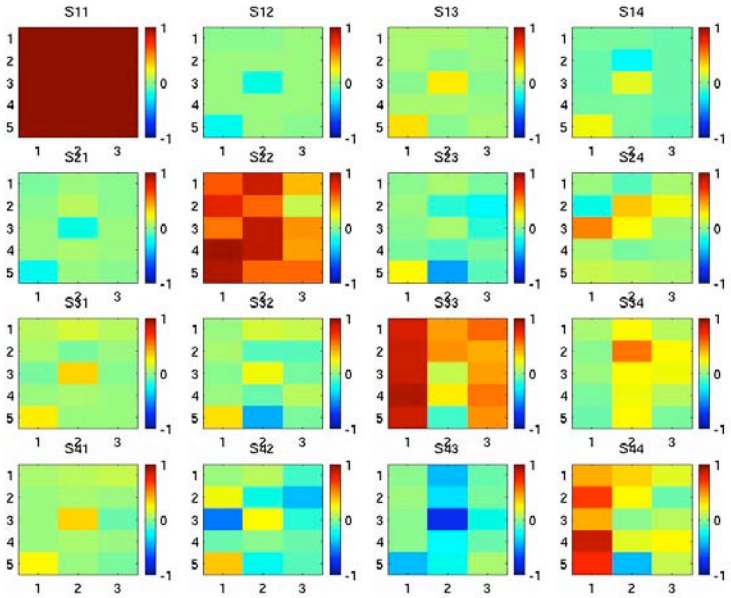

Fig. 5.16. - Mueller matrix for skin sample

MUSCLE

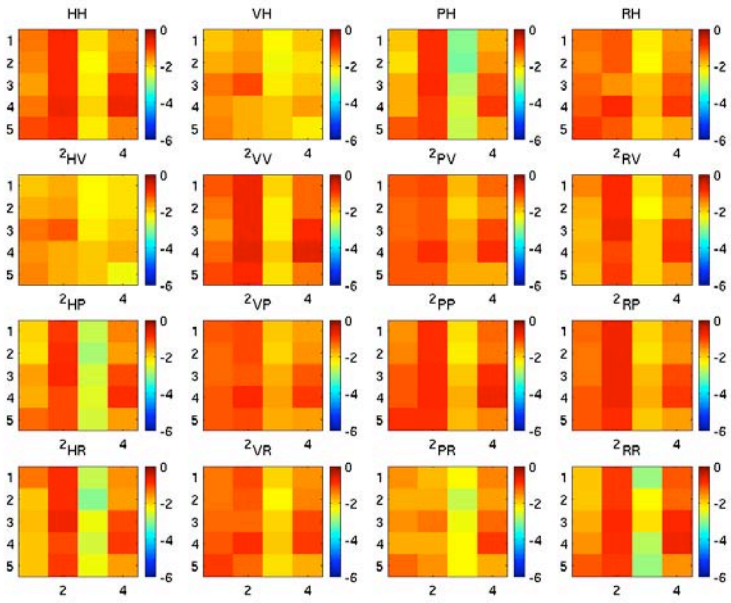

Fig. 5.19. - Data Matrix for 5 positions (y = 1 to 4) on 4 muscle samples of thickness 150, 270, 360 and 420 mm (x = 1 to 4, respectively). Color map = $\log_{10}(T)$.

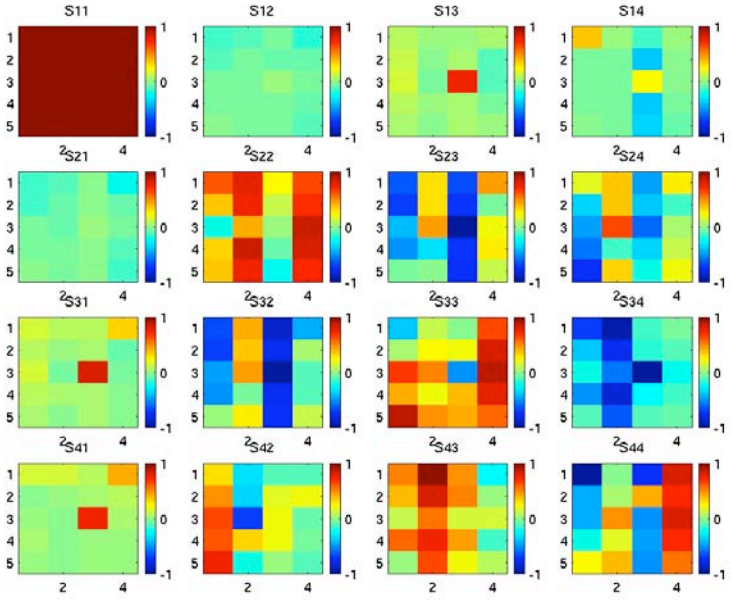

Fig. 5.20 - Mueller matrix for chicken muscle sample.

LIVER

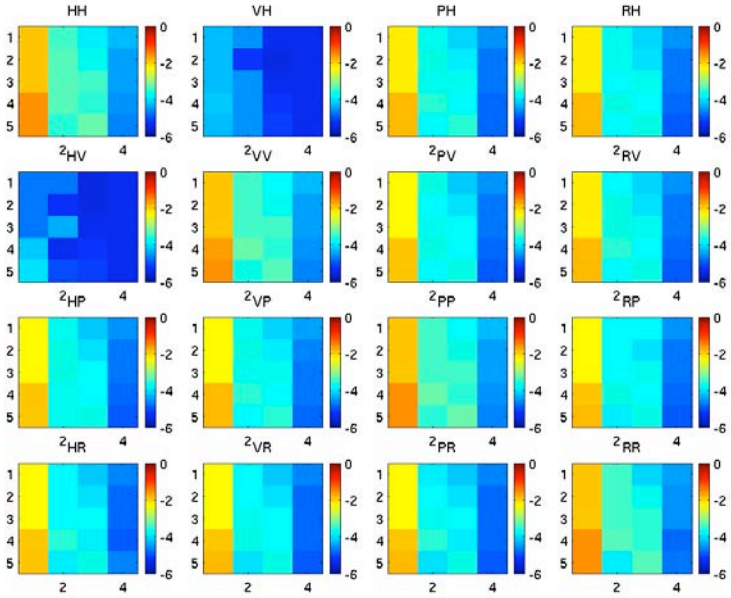

Fig. 5.21 - Data Matrix for 5 positions (y = 1 to 5) on 4 liver samples of thickness 200, 360, 420 and 900 mm (x = 1 to 4, respectively). Color map = $\log_{10}(T)$.

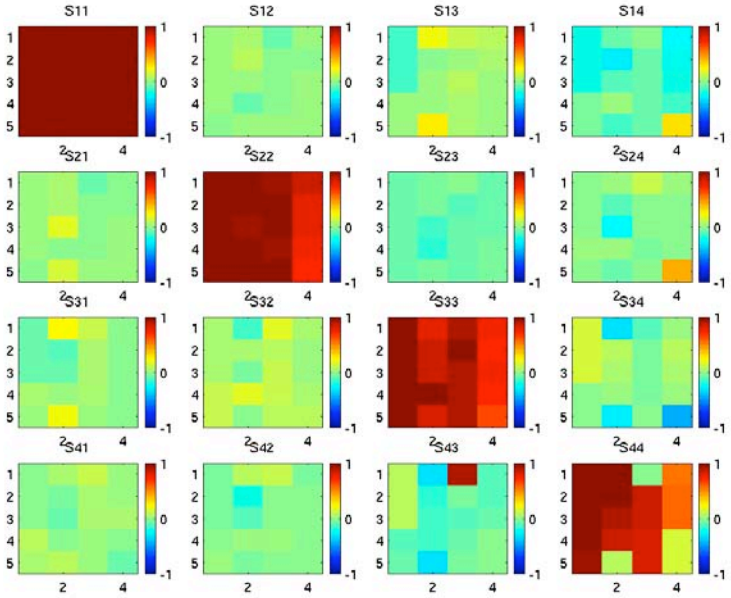

Fig. 5.22. - Mueller matrix for liver sample.

Figure 5.23, 5.24 and 5.26 provide an alternative presentation of the same data. The Data Matrix and Mueller Matrix for the three tissue types, where each element is shown as a graph of the matrix element value versus the thickness (L) of the tissue sample. The Data Matrix illustrates the basic exponential attenuation of transmission as L increases. The Mueller Matrix behavior is characterized by a diagonal elements that transition from a high value to a low value as L increases. In contrast, the other elements remain relatively close to zero. The skin and liver data better follow this summary than do the muscle data which are rather noisy. Nevertheless, this summary is the general behavior.

Note that the normalization of the Mueller Matrix by the MM(1,1) value controls for the general attenuation by the tissue thickness. Therefore, the attenuation of the Mueller Matrix elements MM(2,2), MM(3,3) and MM(4,4) characterizes the loss of the degree of polarization, not simply the loss of intensity. Note that the skin attenuates these elements quite rapidly, dropping toward zero after only a thickness of about 300 mm. In contrast, these elements are only slight attenuated after nearly 1 mm of thickness in liver. The muscle data is somewhat intermediate in its behavior, although the data is too noisy for strong conclusions.

SKIN

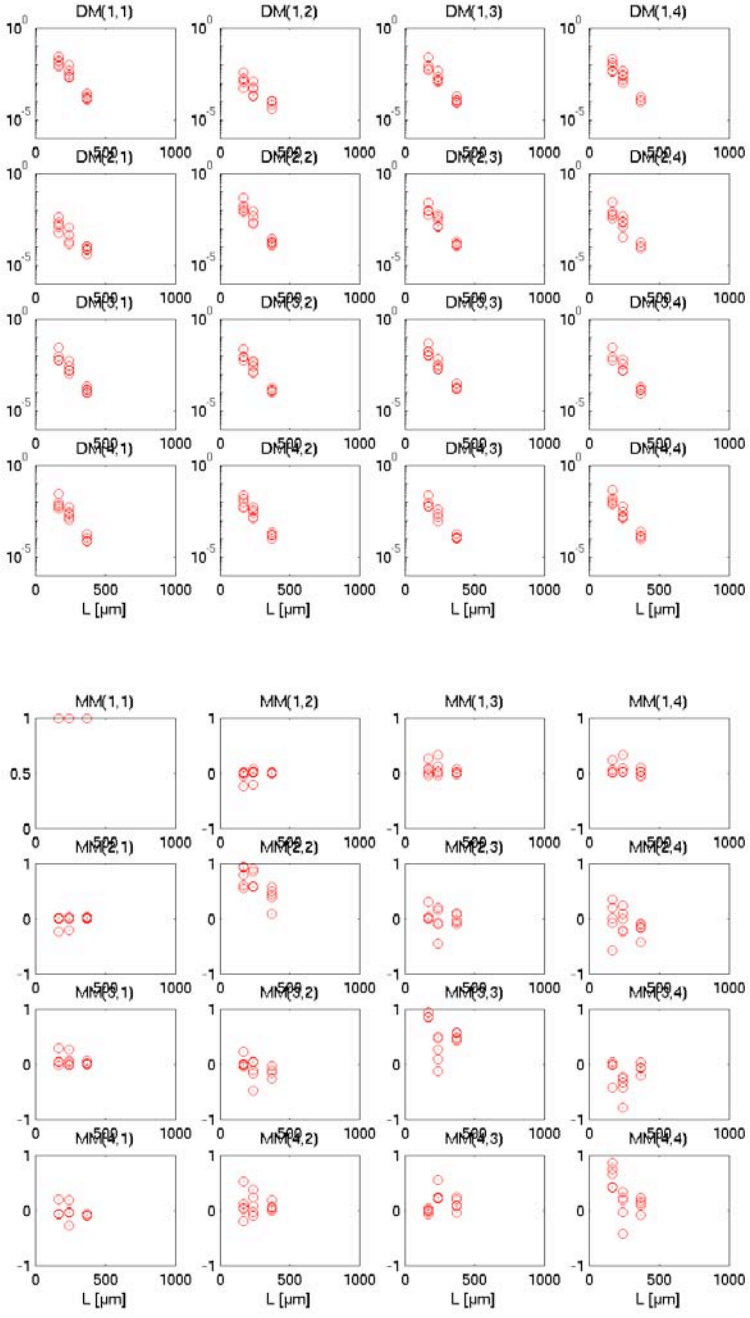

Fig. 5.23. - Skin. (TOP) Data Matrix values versus tissue thickness L (µm). (BOTTOM Mueller Matrix, normalized by MM(1,1), versus L.

MUSCLE

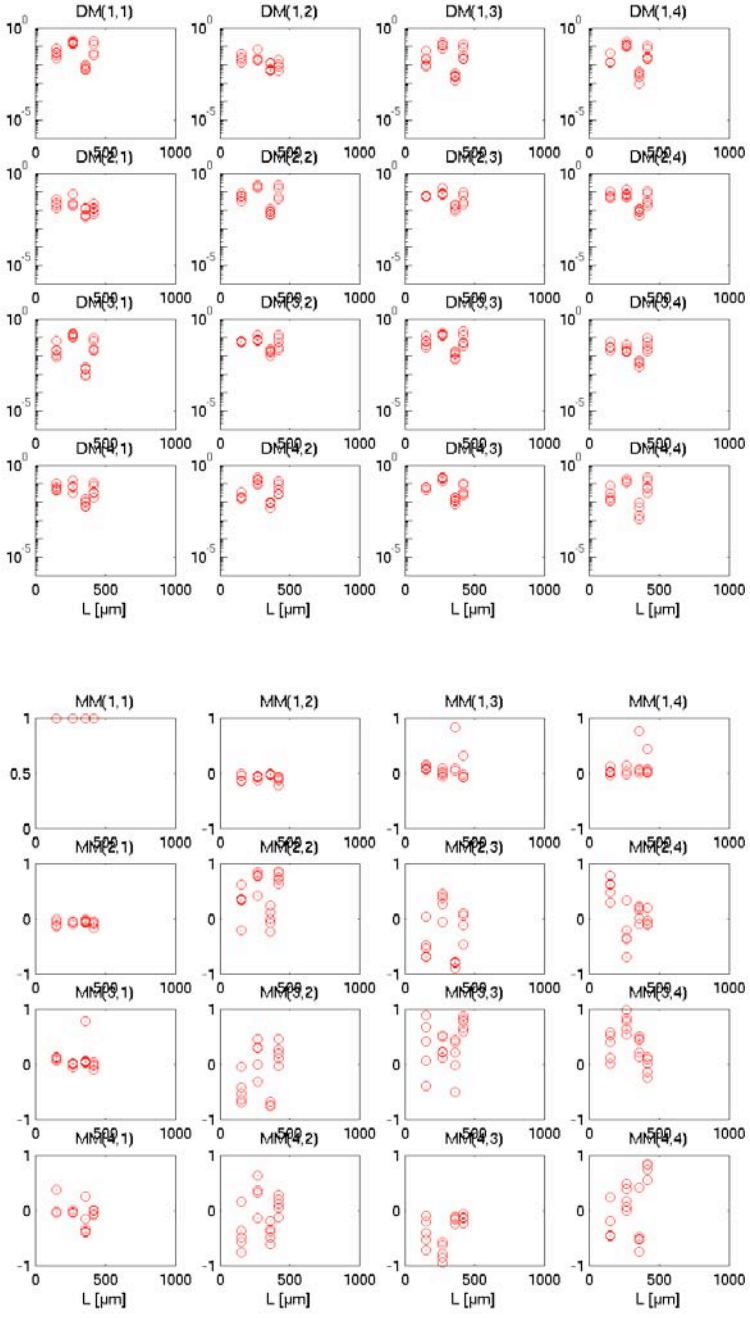

Fig. 5.24. - Muscle. (TOP) Data Matrix values versus tissue thickness L (mm). (BOTTOM Mueller Matrix, normalized by MM(1,1), versus L.

LIVER

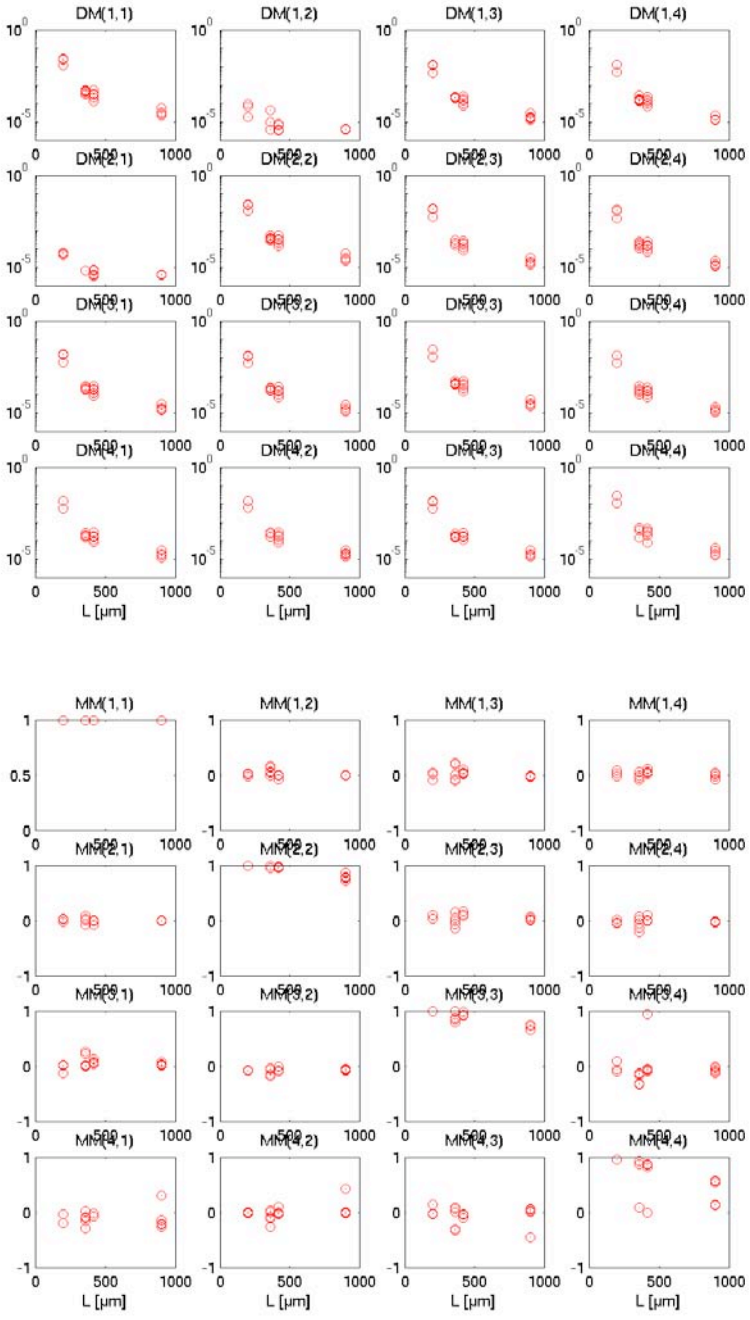

Fig. 5.25. - Liver. (TOP) Data Matrix values versus tissue thickness L (mm). (BOTTOM Mueller Matrix, normalized by MM(1,1), versus L.

Figure 5.24 provides another perspective on the behavior of these tissues. The Total Degree of Polarization (TDP) was calculated by multiply a Stoke vector of purely horizontally linear polarization, S_i = [1 1 0 0]', by the Mueller Matrix of the tissue, MM, to yield an output Stokes vector S_o = [I Q U V]'. Then the TDP was calculated based on S_o.

$$S_o = MM \times S_i$$

$$TDP = \frac{\sqrt{Q^2 + U^2 + V^2}}{I}$$

The plot of TDP versus thickness L is shown in Fig. 5.24. For all three tissues, the IDP falls as a function of tissue thickness L. The rate of attenuation of IDP is skin > muscle > liver. The attenuation of TDP indicates that polarization is not simply being converted from linear into elliptical polarization, but rather there is a true depolarization of the beam occurring.

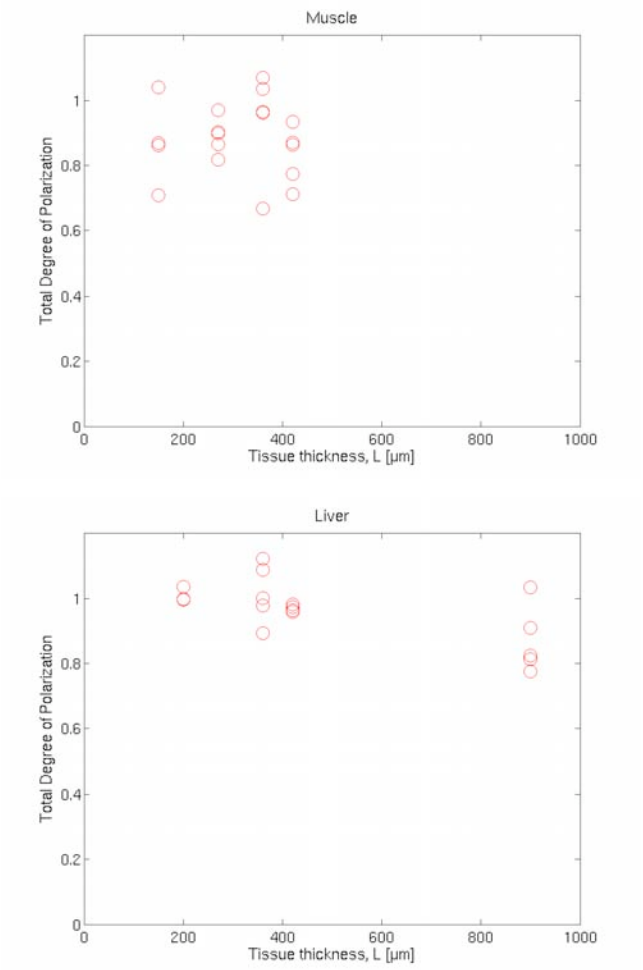

Fig. 5.26. - The loss of Total Degree of Polarization versus tissue thickness (L [mm]) through which the laser beam passes. Tissues are skin, muscle and liver.

5.3.7 Discussion

The ideal retarder model predicts periodic variation in the degrees of linear and circular polarization as a function of tissue thickness, the tissues do not behave as a simple retarder because the degree of polarization falls off with increasing tissue thickness L. A simple retarder would not depolarize the beam, but simply redistribute the polarization over various linear and circular polarization states. The microscopic study of thin tissue sections revealed there are microdomains [90] of birefringence in tissues, on the order of 10-100 μm in size. The orientation of the fast axis of each microdomain is randomly distributed. Hence, a macroscopic laser beam will encounter several microdomains laterally in parallel, as well as encountering several microdomains sequentially as the beam passes through a tissue greater than 100 μm. The loss of polarization is due to the random orientation of microdomains in tissue.

5.4 Characterization of microscopic domains of birefringence in thin tissue sections

The three previous sections showed large variations of polarization properties both inter and intra sample. The variability encountered in the above Mueller matrix experiments could suggest that the laser beam was encountering local domains of birefringence with different orientations and retardance. The ~1mm diameter of the laser beam might be too large and might average over several microdomains. Microscopic domains of iso-birefringence are about 10-100μm. Therefore measurements with a polarization microscope were conducted to characterize the size distribution of such microdomains of birefringence in three 3 tissue types (liver, muscle, skin). In this initial study, the fast-axis orientation and retardance of micro-domains in thin sections of tissue samples are measured.

4.4.1 Materials and methods

Tissue sections of calf liver, chicken breast muscle, and pig skin were prepared by freezing the tissue and then cutting about 200-400 μm thicknesses of tissue. The samples were placed on glass slides with cover-slips and viewed through a polarization microscope. In Figure 5.27 the main elements of a polarized microscope are shown.

Linearly polarized light illuminated the sample from below. A band-pass filters selected a wavelength of illumination equal to 540 nm ±4 nm. The linear polarizer in front of the light source could be rotated manually to generate different states of linear polarization. The analyzing linearly polarizer in front of a CCD camera was in a fixed position.

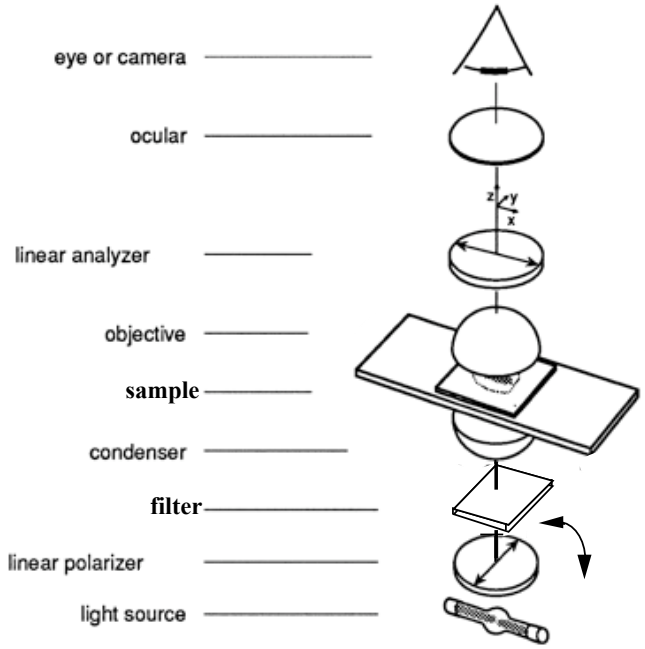

Fig. 5.27 - Schematic of the polarized microscope

The angle θ between the source and analyzer linear polarizers was varied from 0 to 90 degrees in 10 degree interval. In Figure 5.28, the transmission of light at a single pixel is plotted for each image versus angle θ. A fitting routine fit the plot to the function $A\sin^2(\theta) + B\cos^2(\theta) + \text{offset}$

If no sample was present in a microscope the intensity would behave as

$\cos^2\theta$. The considered samples could not be fit to a simple $\cos^2\theta$ (Malus' law), a second $\sin^2\theta$. term was then added.

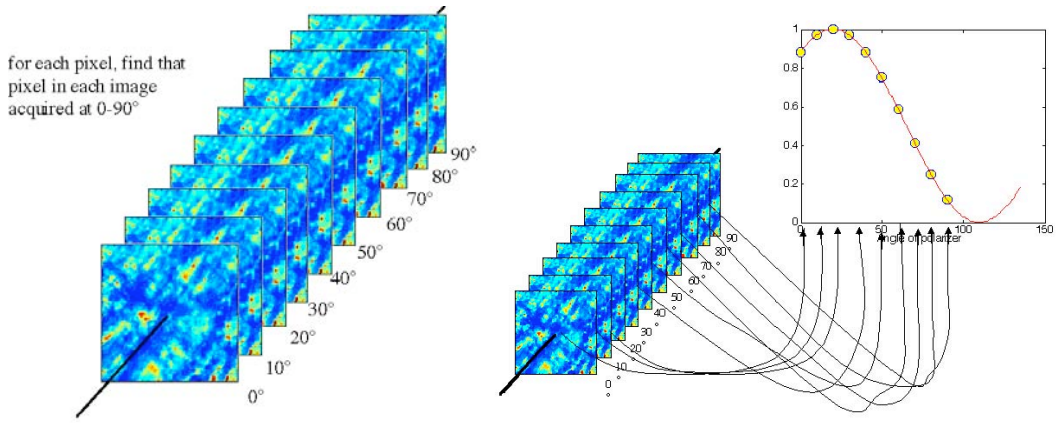

Fig. 5.28 - The ten corresponding pixels in ten transmission images are analyzed versus angle between source and analyzer polarizer.

The shift, $\Delta\theta$, was determined as shown in Fig. 5.27

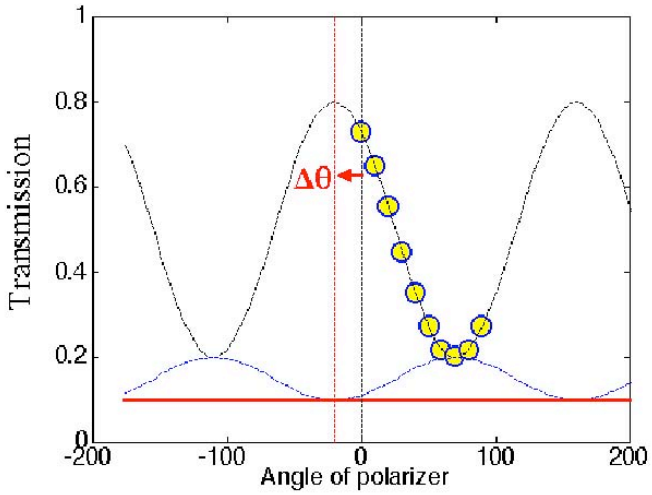

Fig. 5.29 –The transmission versus angle data was fit as $A\sin^2(\theta) + B\cos^2(\theta)$ + offset versus θ. The shift $\Delta\theta$ was determined.

5.4.2 Results

Typical results for muscle samples are shown in Figure 5.29 and 5.30. A transmission image for $\theta = 90°$ is shown on the top. The retardance image is shown on

the bottom. The range of retardance values in the case of muscle varied from zero to 60 degrees. Figure 5.30 shows typical results for liver, muscle, and skin. A visual assessment of the size of iso-retardance domains was on the order of 10-100 μm.

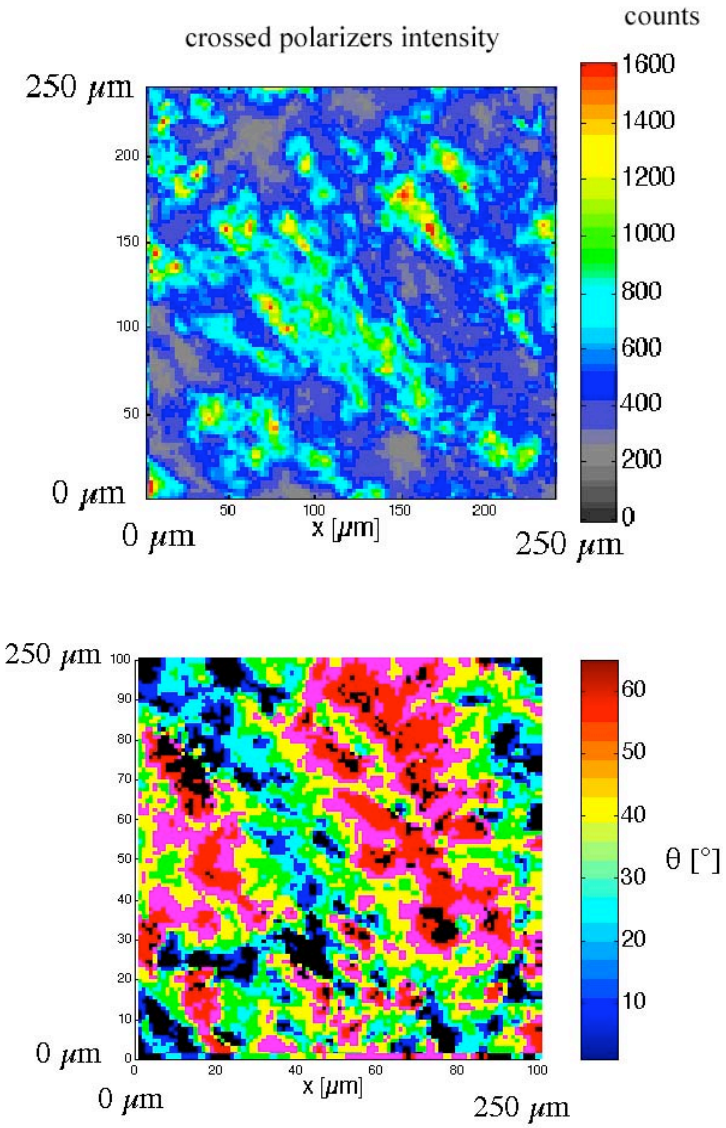

Fig. 5.30 – Muscle sample. (Top) Transmission image for θ =90 (crossed polarizers) . (Bottom). Map of the retardance, Δθ [degrees]. The size of iso-retardance domains is on the order of 10-100 μm.

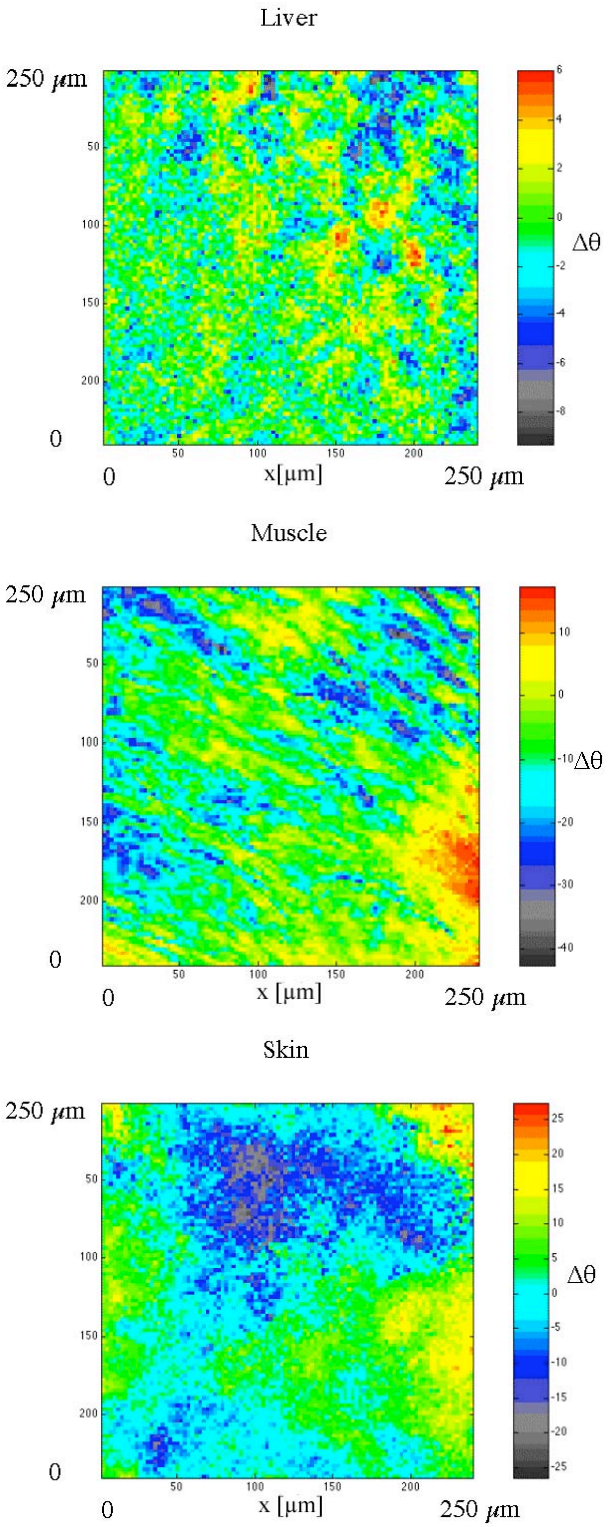

Fig. 5.31 – Maps of the degree of rotation for liver, muscle and skin

Birefringence is reported as the apparent difference in refractive index, Δn , between the fast and slow axes of oriented fibers: $\Delta n = \delta\lambda/(2\pi d)$ where δ is retardation, d is tissue thickness, and λ is the wavelength of light. Values from the literature for tendon are:

Δn	Author	Year	Ref
$3.7 \bullet 10^{-3}$	deBoer	1997	[7]
$3.0 \bullet 10^{-3}$	Maitland	1997	[32]
$3.0 \bullet 10^{-3}$	Stanworth, Naylor	1953	[105]
$4.3 \bullet 10^{-3}$	This thesis	2003	-

Table 5.3 - Comparison in between values of birefringence available in the literature and this thesis values.

In this experiment values of $\Delta\theta$ = 10-20° degrees for a couple hundred μm thickness of pig skin corresponds to a birefringence of $\Delta n = 4.3 \times 10^{-3}$ to 8.6×10^{-3} at 543 nm wavelength.

5.4.2 Conclusions

The size of microdomains of iso-retardance is in the range 10-100 μm, which suggests that optical measurements with laser beams that are on the order of 1-mm in diameter or with imaging cameras with pixels sizes on the order of 100 s of μm will average over several microdomains and consequently complicate interpretation of measurements.

Chapter 6
Evaluation of spherical particle sizes with an asymmetric illumination microscope*

6.1 Introduction

This work describes the first important issue to solve when determining particle size: the determination of a particle size in a monodisperse solution. An inverted microscope and a goniometric assembly that directed monochromatic light onto the scattering particle at different angles was used to this purpose. A cooled CCD camera was used to capture images of scattering patterns from a single latex microsphere. Microspheres of different sizes were measured.

The goal of this technique is to enrich an image of small particles obtained with a high magnification microscope by adding specific information on particle size which are not easily obtainable with a microscope alone. This would be particularly helpful in sizing sub-cellular structures and in detecting changes in cell morphology *in vivo*.

A model was developed to determine the particle size, when the relative index of refraction is known.

---

* This chapter was published in the journal "Selected topics in Quantum Electronics" Vol 9. n.2 301-307 March/April 2003

A polarized microscope system is used to perform goniometric measurements of light scattered by small particles. The incident angle light on a sample of monodispersed latex microspheres is changed and a microscope objective lens collects scattered light from the samples. Light is only collected at angles greater than the objective lens numerical aperture, so that only light scattered by the spheres is collected. The experimental results were modeled with a Mie theory-based algorithm. Experiments conducted with microspheres of diameter 1.03 μm, 2.03 μm and 6.4 μm show that, by decreasing the objective lens numerical aperture from 0.55 to 0.0548, a more distinguishable scattering pattern is detectable. From these highly shaped curves, I found that the size of a sphere of nominal diameter 2.03 μm was 2.11 ± 0.06 μm and a 6.4 μm sphere was 6.34 ± 0.07 μm.

Morphological changes in cell nuclei can indicate a pre-cancerous state, and for this reason many studies have tried to characterize cell nuclei as means of early cancer diagnosis [40,41,42,43]. Backmann and Perelman [25] [26] developed a polarized light spectroscopic method to quantitatively measure epithelial cell structure *in situ*, and in particular the nuclear size and relative refractive index. They distinguished healthy from cancerous mucosal tissue by determining the size distribution of epithelial cell nuclei: cancerous epithelial nuclei are dysplastic and larger than normal nuclei. Solokov *et al.* [40] used similar techniques to measure cell nuclei sizes and indices of refraction. These spectroscopic techniques could lead to non-invasive early detection of epithelial cancer, which constitutes 90% of all cancers. Canpolat and Mourant [35] developed a fiber optic probe for particle size analysis in turbid media. They used a single fiber for delivery and collection and matched their results to Mie theory and Monte Carlo models.

In microscopy, Ovryn and Khaydarov [102,103] developed a method to assess the location of the scattering spheres in three-dimensional flows. They spatially resolved the detailed structure of the scattered light from spherical particles in tenuous media, using a high numerical aperture microscope and a single wavelength. They were able to apply their model to the measurement of cell nuclei size using a reference library of scattering

patterns from known particles, matching the theoretical patterns to experimental ones with a neural network.

Asymmetric illumination contrast (AIC) has been used in microscopy [104,105] in the past to enhance image contrast, but was largely abandoned after the invention of phase contrast microscopy. More recently, AIC has been used for three-dimensional imaging of cellular structures [106]. Our method differs from AIC by using only angles of incidence above the objective lens numerical aperture (N.A.), so that unscattered light is not collected, and only light scattered by the particle is collected. Unlike AIC, our goal was not to enhance contrast, but to use the angular information to build a particle phase function. I used this method to measure the size of latex microspheres ranging from 1.03 to 6.4 micrometers.

6.2 Materials and Methods

An inverted microscope (Eclipse TS100, Nikon, Mellville, NY) was modified by replacing the original light source with a goniometric assembly to allow illumination at arbitrary angles. A 16-bit CCD camera (Princeton Scientific, Trenton, NJ) was mounted on the microscope to collect high-resolution images.

The goniometric light source assembly is shown in Figures 6.1 and 6.2. Light was delivered from a 630 nm dye laser, pumped to the microscope by a frequency-doubled Nd-YAG laser (Laserscope, San Jose, CA) through an optical fiber with a core diameter of 0.6 mm. To average speckle and improve image quality, the optical fiber was shaken with an aquarium pump. The incident angle of the beam was adjusted by positioning the fiber at different angles. The considered angles range for θ_c was 33.7˚ to 80˚ degrees ($\sin^{-1}$ NA = 33.7 degrees). To center the beam in the sample plane for every incident angle, the optical fiber was mounted on a rotating arm. A micro-lens assembly collimated the light beam; the spot size was approximately 4 mm in diameter. The sample was composed of latex microspheres (Ted Pella, Redding, CA, Duke Scientific, Palo Alto, CA, Bangs Laboratories, Fishers, IN) in a diluted aqueous solution, sandwiched

between glass slides and positioned, with glass supports, in the center of the pivotal plane. The index of refraction of the microsphere was 1.59 and the index of refraction of the aqueous solution was 1.33. Angles were measured with a protractor.

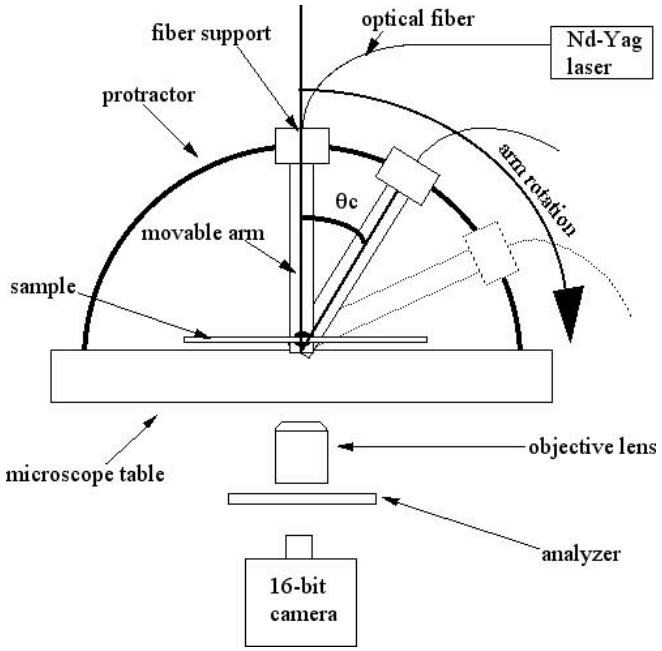

Fig. 6.1. - Front view of the goniometric assembly. The optical fiber rotates around a pivotal point; the fiber is mounted on a Delrin arm. The exact angular position of the incident beam is measured with a protractor.

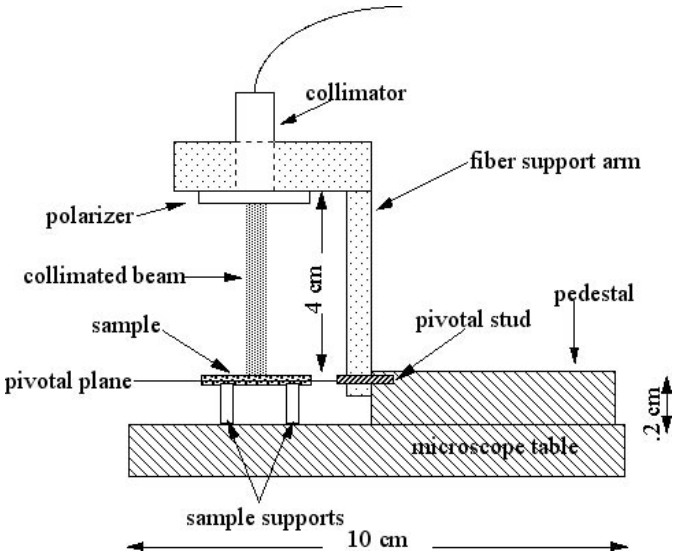

Fig. 6.2. - Detail of the fiber support arm. The laser spot is centered on the pivotal plane for every θ angle of the support arm. The sample is maintained at the center of the pivotal plane with glass supports. A pedestal connected to the microscope table with double-sided tape assures the rigidity of the system.

The incident light was linearly polarized with a polarizer (Hinds, Portland, OR). An analyzer, positioned behind the microscope objective lens, selected only one polarization state. The extinction ratio of the polarizer analyzer pair was 1:10000. During experiments the polarizer and analyzer were either both parallel or both perpendicular to the plane of incidence. For every arm position, IPlab^{TM} software captured an image of the sample. Acquisition time was one second. The objective lens used for all the experiments was a 40x magnification objective with 0.55 N.A.; the working distance of the objective lens was 2.1 mm.

6.2.1 Image evaluation

For each particle size a minimum of one image for every five degrees was captured. All images were analyzed with Matlab^{TM} software. To account for the camera's intrinsic dark noise, a dark noise value of ~460 counts was subtracted from every image. The individual microspheres stood out in the captured images as bright

round circles. For non-perpendicular illumination, the microspheres exhibited this circular shape, but with a bright tail; this tail was not considered during the analysis. To analyze the images, I sampled a region of interest (ROI) of ±4 pixels around the center (X_o, Y_o) of the sphere's circular image. The mean of the ROI was calculated and plotted versus the incident angle θ. This process was repeated for three spheres on the same image.

6.2.2. Collection model

A simple model based on Mie scattering was implemented. The model incorporated the numerical aperture of the objective lens so its ability to collect light-rays scattered by the microsphere was accurately simulated. Figure 6.3 shows the geometry of the collection model. θ_i defines the incident ray's angle relative to the normal to the glass slide. β and ϕ are, respectively, the deviation angle and azimuthal angle of scattering by the microspheres. An objective lens with an aperture specifies a solid angle of collection, θ_{coll}. All the photons scattered within the cone of collection of the objective lens are collected by the objective lens. The deviation angle β depends on the angles θ, θ_i, and ϕ

$$\beta = \cos^{-1}(-\sin\theta_i \sin\theta\cos\phi + \cos\theta_i \cos\theta) \qquad (6.1)$$

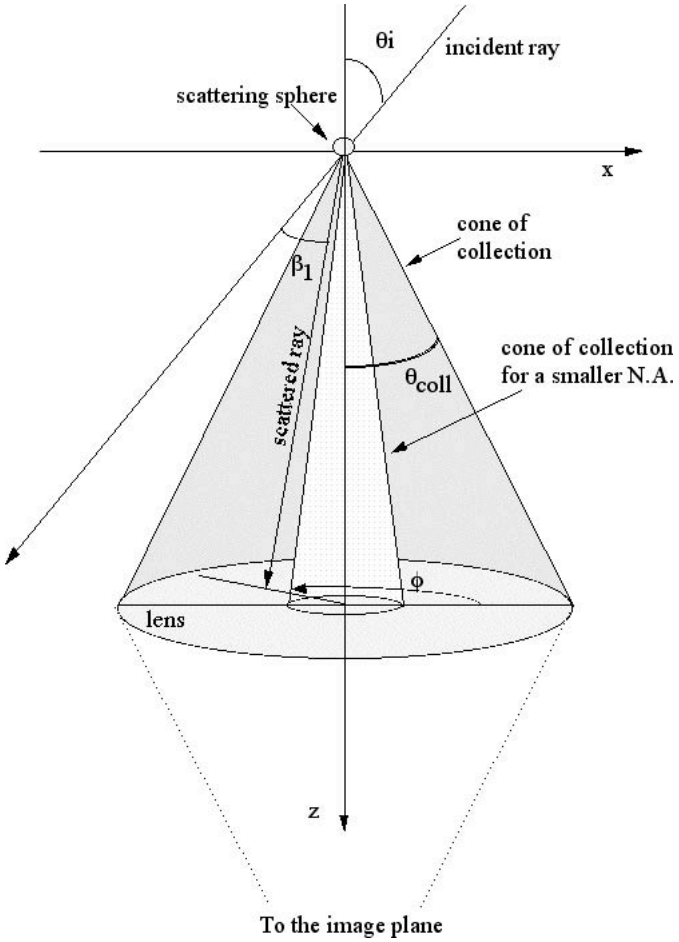

Fig. 6.3. - Collection model showing the definition of angles used. The numerical aperture of the objective lens is defined as N.A. = n sin (θ). The direction of propagation is in the positive z-direction.

Rays scattering from a single particle distribute themselves according to a phase function p(β,ϕ), determined by the particle size and predicted by Mie theory. The light contribution at a particular point depends on both the scattering angle θ and the rotation azimuth angle ϕ. The Stokes vector formalism was used to represent polarization. The incident Stokes vector S=[I, Q, U, V] is projected into the scattering plane using a rotational matrix **R**(ϕ) (see [10]). The scattering at an angle θ is regulated by a scattering matrix **M**(θ) whose elements are given by Mie theory. The **S** vector is rotated – ϕ to evaluate it in the original frame of reference. To account for the analyzer orientation, a polarizer Mueller Matrix P is added to the equation 6.1.

These steps are conducted for every incident angle θ_i and $0 < \phi < 2\pi$, where the scattered angle β is given by Equation 6.1. The net polarization at a point is given by integrating the scattered Stokes vector over all contributing angles. The Stokes vector is

$$\boldsymbol{A} = \int_0^{2\pi} \int_0^{\theta_c} \boldsymbol{P}\,\boldsymbol{R}(-\phi)\boldsymbol{M}(\beta)\boldsymbol{R}(\phi)\;\; \boldsymbol{S}\; d\theta\;\; d\phi \tag{6.2}$$

The first term of the resulting Stokes vector **A**(1) is the intensity.

Figure 6.4 shows the predicted behavior for a sphere of diameter 2 μm for different numerical apertures of the collecting objective lens. The typical periodicity, or hilly behavior, of the angular scattering curve of Mie theory is noticeably lost as the N.A. increases (0.0548, 0.1, 0.3, and 0.55). The peaks and valleys average to an almost shapeless form for N.A = 0.55, which was the numerical aperture of the objective lens used in the initial experiments.

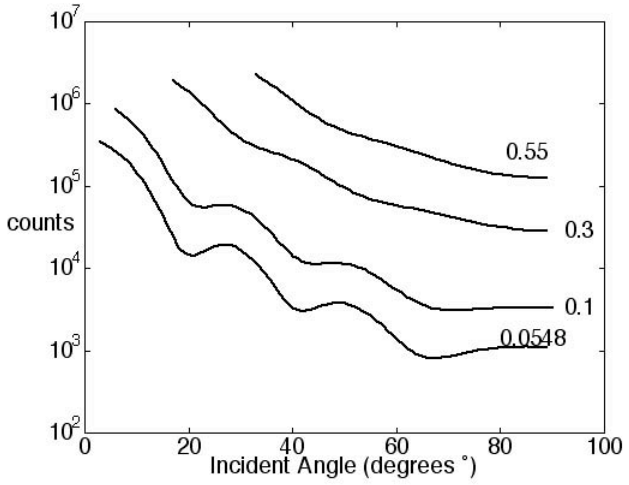

Fig.6.4. - Modeled behavior of perpendicular polarized incident light and perpendicular collection by a microscope objective lens for various numerical aperture for spheres of diameter = 2.03 μm. The plotted numerical apertures are 0.0548, 0.1,0.3, and 0.55. The spheres had an index of refraction of 1.59 and were in a medium with index of refraction 1.33

To obtain a better Mie scattering signature, the N.A. of the collecting objective lens was decreased by restricting the collection area of the lens: a small aperture of diameter D was introduced in front of the pupil of the objective lens. The new effective N.A is given by the equation:

$$NA = \frac{D}{2\ f}$$

where f is the lens focal length. Our aperture was positioned on the front surface of the lens. Restricting the N.A. of the objective lens diminished the image resolution, as predicted by the Rayleigh criterion. Despite these artifacts, single isolated spheres were distinguishable so that the angular information could be evaluated. Three effective N.A were tested: 0.55, 0.0952 (D = 400 μm), and 0.0548 (D = 230 μm).

6.3. Results

Figures 6.5, 6.6, 6.7, and 6.8 show experimental results (dark circles) and the model fit (dark line). The model was scaled with a simple multiplicative scale factor (f_s) to fit the data. Data are expressed in counts. The data start at the critical angle and end at 80 degrees. Values below the critical angle, in which the unscattered incident beam light contributes to the image, were not considered.

Figure 6.5 shows the results of experiments for two sizes of spheres, using a 0.55 numerical aperture. For both experiments, I polarized the source and oriented the detector analyzer perpendicular to the scattering plane. As predicted by the model, the typical Mie scattering shapes averaged out, and I obtained only a smooth exponential decay.

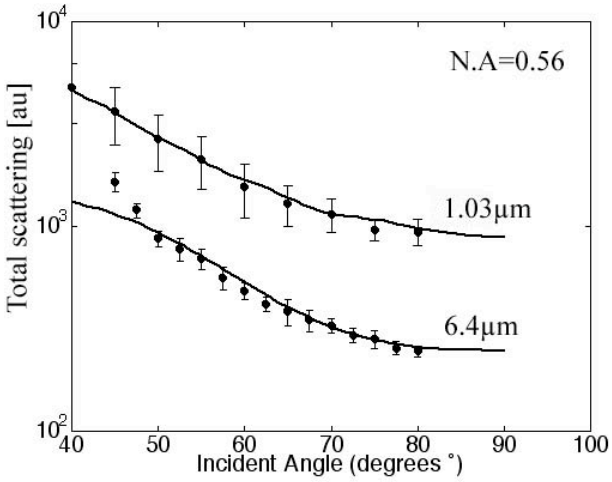

Fig. 6.5. - Perpendicular polarized light scattered by a sphere of diameter 6.4 μm and a sphere of diameter 1.03 μm. The N.A. of the objective lens was 0.56. Scaling factor f_s = 2.5 for the 6 μm sphere and 1.8 for the 1 μm sphere.

Figure 6.6 shows the results obtained with a pinhole diameter equal to 400 μm. In this case, the polarizer and analyzer were oriented parallel to the scattering plane.

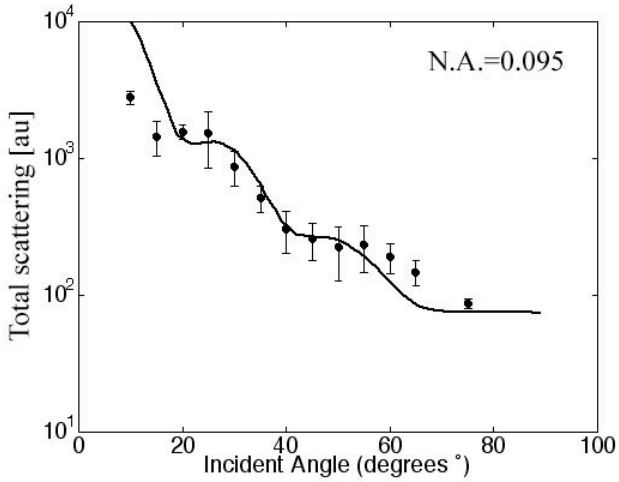

Fig. 6.6. - Parallel polarized light scattered by one sphere of diameter 2.03 μm. Numerical aperture is 0.095. Scaling factor $f_s = 0.1$

Figures 6.7 and 6.8 show results for an aperture of diameter 230 μm and spheres of diameter 2.03 μm and 6.4 μm. More periodic behavior is seen. The variation from the model could be explained in part by an inaccuracy in the measurement of the lens pupil diameter. The real objective diameter might, in fact, be larger than that established with the pinhole diameter. The real objective N.A. would then be larger than assumed.

The first three points of Figure 6.7 and the first seven points of Figure 6.8 are lower than expected because of camera saturation.

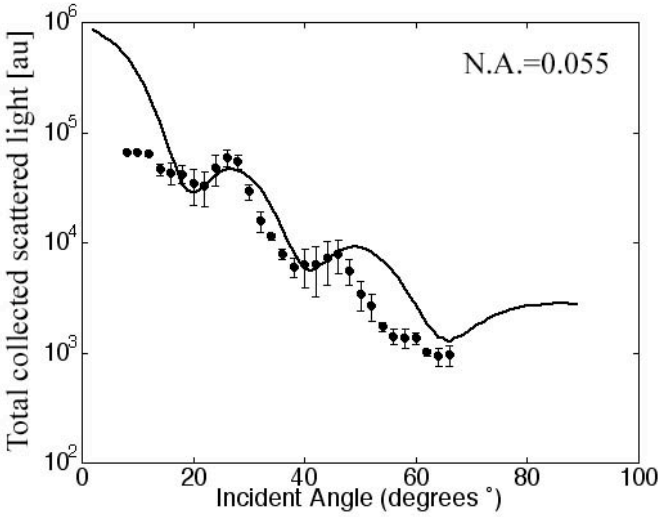

Fig. 6.7. - Perpendicular polarized light scattered by one sphere of diameter 2.03 μm. Aperture 230 μm and N.A.=0.055. Scaling factor fs = 2.1.

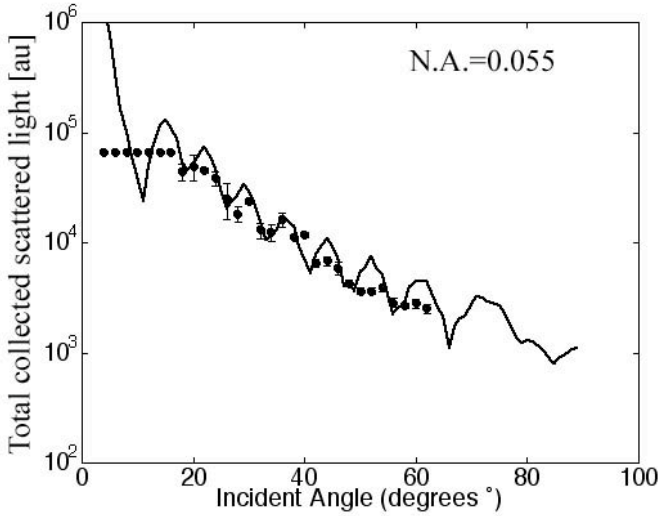

Fig. 6.8. - Perpendicular polarized light scattered one by sphere of diameter 6.4 μm. Aperture 230 μm and N.A.=0.055. Scaling factor fs = 1.1.

6.4 Fitting of the data

All data sets were fitted using a least squares fitting. To do the fitting multiple curves were first generated with our model. The curves were generate using different particle sizes, in increment of 0.01 μm. The model N.A. and index of refraction was the same as in the experiment. The model and the experimental data were normalized by their respective value at θ_c + 10 degrees, 10 degrees were added to avoid imaging artifacts, for the experimental values at angles close to θ_c. The total error between the model and experiment was plotted as a function of the particle size. For every size three different experimental data sets were fitted. A typical result for an experiment conducted with 2.03 μm spheres and N.A. 0.055 is shown in Figure 6.9. A minimum is clearly observable close to 2 μm.

The mean and standard deviation sphere size was calculated by using the size corresponding to minimum error for measurement on three different particle of the same size. For 2.03 μm spheres the mean calculated radius was 2.11 μm with a standard deviation of 0.06μm.

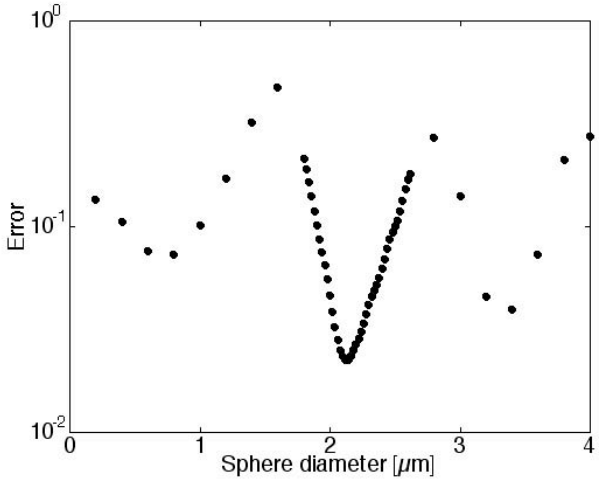

Fig. 6.9. – Total error in total scattering for 2.03 μm diameter microspheres. The size with the minimum used to establish the sphere size. In this case 2.11 μm. Two other minima are visible in the Figure; this could constitute a problem when trying to determine the particle size.

Similar fitting implemented for the 6.4 μm microsphere gave a mean diameter of 6.34 μm and a standard deviation of 0.07 μm. Going to larger N.A. increased the error in the fit as expected. Fitting a 2.03 μm with a N.A=0.095 yields two minima of equal importance: at 2.96 μm ± 0.003 μm and one at 2.58 μm ± 0.03 μm. Finally for the 1.03 μm and 6 μm spheres with a N.A.=0.55 no clear minimum was visible.

Because one application of this technique could be sizing epithelial cell nuclei as they enter a pre-cancerous state, behavior of spheres whose indices of refraction were close to the nuclear index of refraction 1.43 [38], in a medium with index 1.37 using an objective lens with N.A.=0.055 were modeled. Epithelial cells are typically 5–10 μm in diameter [107]. The model showed a clear difference between the predicted behavior for 5 μm-diameter spheres and 15 μm-diameter spheres (See Figure 6.10).

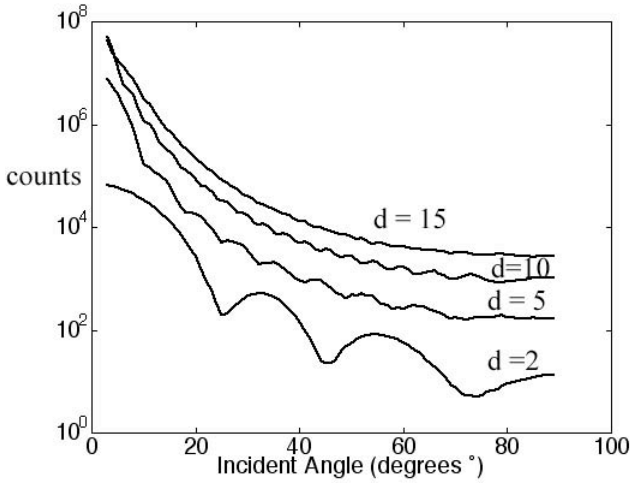

Fig. 6.10. - Modeled behavior of perpendicular polarized light scattering from a modeled cell. The relative index of refraction was 1.036. The nucleus size varied from 2 to 15 microns. Because epithelial cell nuclei range from 5 to 10 μm in diameter, I can expect to be able to detect nucleus enlargement and scattering from smaller particulates such as organelles.

6.5 Preliminary cell work

Some experiments on muscle cells in culture for light incident at 30˚ 40˚ and 50˚ are shown in Figure 6.11,6.12 and 6.13. The nucleus is visible in the large stretched cell. Cell were grown on glass slides in an incubator and brought to the asymmetric microscope setup seconds before the experiment.

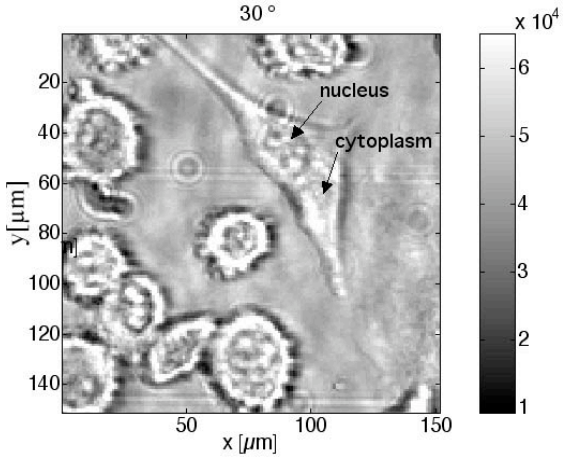

Fig. 6.11 – Images of polarized light scattering from muscle cells when incident light is at 30˚ respect to the sample normal. Nucleus and cytoplasm are visible for the larger cell stretching across the image. This image is very bright with intensity levels around 50,000 counts.

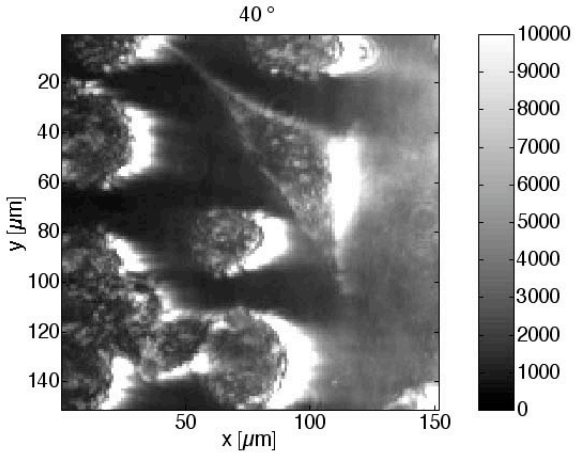

Fig. 6.12 – Images of polarized light scattering from muscle cells when incident light is at 40˚ respect to the sample normal. Some structures inside the cell appear brighter than the surrounding area. In this image the intensity has dropped to 9000 counts, less and less direct light is entering the aperture of the objective lens.

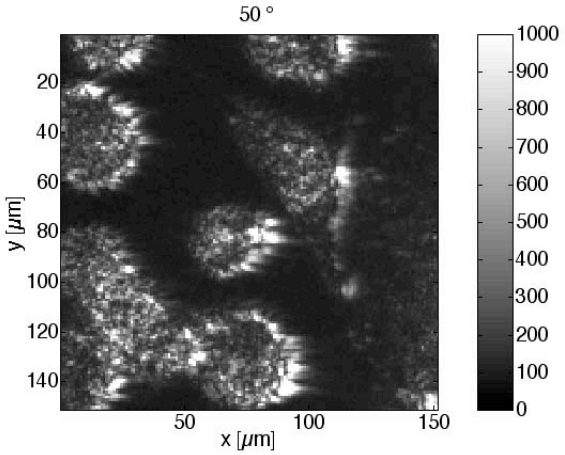

Fig. 6.13 – Images of polarized light scattering from muscle cells when incident light is at 50˚ respect to the sample normal. Some structures inside the cell appear brighter than the surrounding area. In this image the intensity level dropped to less than 1000, only light whose original direction was deviated by scattering enters the objective lens.

Figure 6.13 shows a plot of pixels collected in the nuclei and cytoplasm area of the big triangular shape muscle cells versus the incident angle. Cytoplasm and nucleus pixel have different scattering intensities and probably shape.

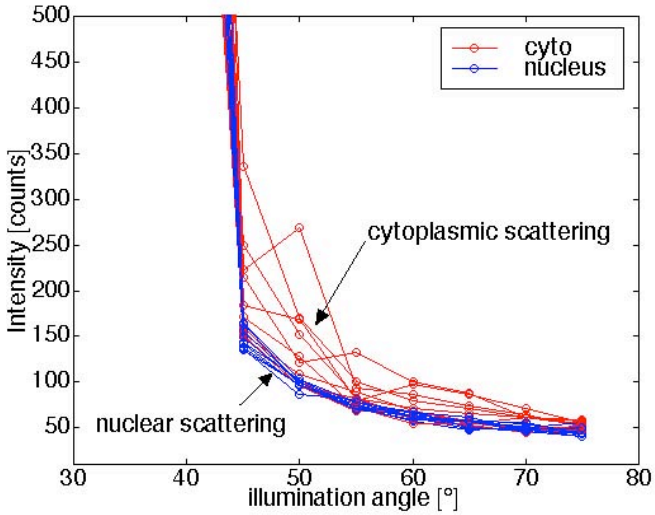

Fig. 6.14 - Graph of scattered intensity versus the illumination angle for pixels within the nucleus and the cytoplasm of the triangularly shaped muscle cell. Cytoplasm scatters light more broadly than the nucleus.

This is a very early although encouraging result that suggests that this technique could enhance the discrimination between nuclei and cytoplasm

6.5 Discussion

A new method to evaluate a sphere particle size was introduced using an inverted microscope and a goniometric assembly to illuminate a scattering sample at various incident angles. The behavior of a monodisperse solution of particles was modeled, and a range of particle going from 1.03 μm to 6.4 μm in diameter were analyzed. The first important element in this kind of measurements resides in collecting only light that has scattered from the sphere thus avoiding the unscattered light that has no valuable information. To do so measurements only with the light source oriented at angles greater than the numerical aperture of the objective lens were performed.

When a microscope objective lens of 40x and a N.A. = 0.55 was used in our measurement I obtained curves that were smooth and did not offer any particular shape apart from an exponential decay, so a sphere of 1.03 μm in diameter was easily confused with a sphere six times bigger. This was due to the fact that in the image I was averaging a large range of scattered rays, the only limitation being the cone of collection of the objective lens. The hilly shapes typical of certain sizes for the specific wavelength were averaged out to generate the smooth curves of Figure 6.6. The solution to this problem was to decrease the objective lens cone of collection while trying not to sacrifice visibility. When the N.A. was reduced to 0.055, the cone of collection decreased of 10 times allowing a better selection of the scattered rays. The curve experimentally obtained with such an aperture showed allowed a better discrimination between different sizes.

A least squares fit of the experimental data was generated and only the results obtained with the small objective lens numerical aperture converged to the true size values. In the example proposed in Figure 6.9 for a 2.03 μm sphere, the error of the fit showed an absolute minimum at 2.11 μm. Other relative minima were visible in the error curve at 3.3 μm and at 0.7 μm. This could constitute a problem when no information of particle size is available *a priori*. A more refined minimization algorithm that includes the information of the number of relative maxima present in the model curve compared to the number of maxima present in the experimental curve, may drastically improve the size identification.

In this section monodisperse solution of spheres were considered, but given the narrow cone of collection an optical fingerprint of the dominating particle size could be extracted also from polydisperse solutions.

One of the potential uses of this technique is to identify morphological changes in cell nuclei. The advantage of this technique is in the simultaneously imaging of the cells and quantitative analysis of cells size. A very preliminary test was conducted on a culture of smooth muscle cells that showed an encouraging discrimination in between scattering from cytoplasmic areas and nuclear areas.

Chapter 7
General discussion and conclusions

7.1. Introduction

This dissertation has presented instrumentation and theoretical studies on polarized light imaging for the detection of skin cancer borders. The proposed instrumentation is an inexpensive hand-held device that may acquire polarized images at 7 frames a second. Images may be saved in pict format for later analysis. *In vivo* measurements were performed on patients undergoing Mohs surgery or other surgery on a variety of lesions. 10 patients were tested.

Monte Carlo models were designed to understand polarized light propagation. Three conceptually different models were introduced yielding the same results. I used these models to improve the design of instrumentation and to test an early heuristic model of polarized light propagation in scattering media. The models also enable a better understanding of the dominant factors in transmission mode experiments (effect of anisotropy g in Figure 4.23).

Polarized light penetration depth and bulk retardance was measured *in vitro* for different biological tissues to further study the effect of the various tissues structural components, such as actin myosin in muscle, and collagen in skin. Microscopic areas of iso-birefringence were analyzed with a polarized light microscope, and a novel asymmetric illumination microscope was introduced as a method to determine nuclei swelling in precancerous cells. Multiple tests on micro spheres of different sizes showed

that this method can correctly predict the particle size once some parameters, such as particle index mismatch and incident wavelength, are known. The technique was applied *in vitro* on muscle cells.

7.2. Construction of a polarized hand-held device for skin cancer borders determination

A unique hand-held polarized light camera was built to give doctors at the OHSU dermatology department the ability to quickly collect images of skin lesions. During the construction the importance of image resolution and registration was realized. Therefore, small field of view (area < 0.25 mm^2 per 160 10^3 pixels) was used, and gave very good images comparable to the one obtained with the 16 bit digital camera. The necessity to go to larger image size (2.25 cm^2) reduced image resolution, and the collected images were not as interesting as previously collected images, although they gave a more general idea of cancer borders. Another fundamental parameter to correct was the image registration; the two CCD cameras were mounted on both sides of a beamsplitter cube on a supporting delrin cube, this construction allowed a coarse registration of the Parallel and Perpendicular images. It was soon realized that a finer (< than 4 pixels difference) registration must be implemented in the software on both axial directions and for skew. The images collected in the clinic showed some surprising facts. The first ever published polarized light images of melanoma *in situ* showed a very strong contrast in the pol images with visible collagen wells surrounded by melanotic areas. Images of superficial capillary as well as skin graft and skin burn were collected and showed significant difference with respect to the image collected with normal unpolarized light; this could mean that this technique could be beneficial even in other fields, such as portwine stain assessment before and after laser treatment.

This camera is the predecessor of much better and equally inexpensive devices; in fact web cameras, digital cameras and digital camcorders quality is increasing rapidly while costs seem to drop at an even higher pace. An error analysis of this system showed

that web cameras with lower signal to noise would be preferable. A more in depth analysis of the different camera settings should also be attempted in order to find the optimal signal to noise ratio. Finally I think that this camera could work for detection of borders of cervical cancer. Preliminary experiments of a cow uterus wall showed very structures system of collagen bundles. The disruption of these bundles could be accentuated polarized light imaging.

7.3. Three Monte Carlo programs of polarized light transfer into scattering media

Constructing Monte Carlo programs to evaluate polarized light transport into scattering media was necessary since no program of this kind is available as shareware or commercially. Initial evaluations herein of the depth of penetration of polarized light into biological media and gel phantoms indicated some experimental findings that seemed to contradict the current literature. In particular it was discovered that the depolarization of larger spheres in transmission mode (Figure 4.4) was more rapid than previously thought, and that this phenomenon does not depend uniquely on the size parameter x, but also on the anisotropy g. With the new Monte Carlo program there is the possibility to model the experiment and show the anisotropy influence; for example the anisotropy g for 2μm spheres is smaller than g for 1μm spheres, monodisperse solutions of 2μm spheres will depolarize faster that 1μm spheres. Until now it was generally accepted that smaller spheres depolarized light faster than larger spheres.

The first Monte Carlo program made was based on the Bartel and Hielsher [42] method. During the construction of this model it was realized that the available literature was rather cryptic and targeted to an audience of astrophysicists and meteorologists. It was then decided that it would be of great interest to the biomedical community to provide a restatement of the method behind polarized light Monte Carlo. Following an extensive literature search, two different methods dominated the writing of this kind of Monte Carlo code: the meridian plane method and the Euler rotations methods. It was also realized that this latter method could be improved with the use of quaternions. Once

the programs were written extensive testing was performed, comparing the generated results with the results offered by Evans [65] in his adding doubling code and experiments. The comparison of Monte Carlo program to Evans showed that his interpretation of flux could not be applied to the experiments herein since he employed a local detector for collection. During the experimental testing it was realized that the back-reflectance Mueller matrices from monodisperse solutions of micro spheres could be obtained with a program that had substantial errors, such as a wrong rejection method ,or wrong final rotations to the laboratory frame of reference. This is caused in part by the fact that many of the 16 images that constitute the Mueller matrix are axially symmetrical. The real test was to break symmetry. This was done both experimentally and with Monte Carlo programs, resulting in finding more errors in my programs. After corrections the programs were used to simulate a variety of asymmetrical experiments such as in Figure 3.6 and Figure 3.11. All three programs will be available for download to the general public. Moreover, chapter 4 was submitted for publication and will hopefully aid the field of biomedical optics in the better understanding of this complex subject.

In the future I hope this model will be used as standard, just like the original Monte Carlo program proposed by Liong Wang and Steve Jacques. The next step for these Monte Carlo programs is the implementation of boundaries using Mueller matrices for Fresnel reflectance and transmission.

An interesting use of these programs would be the implementation of layers of different sphere sizes, mimicking the behavior of skin layers as in Figure 7.1

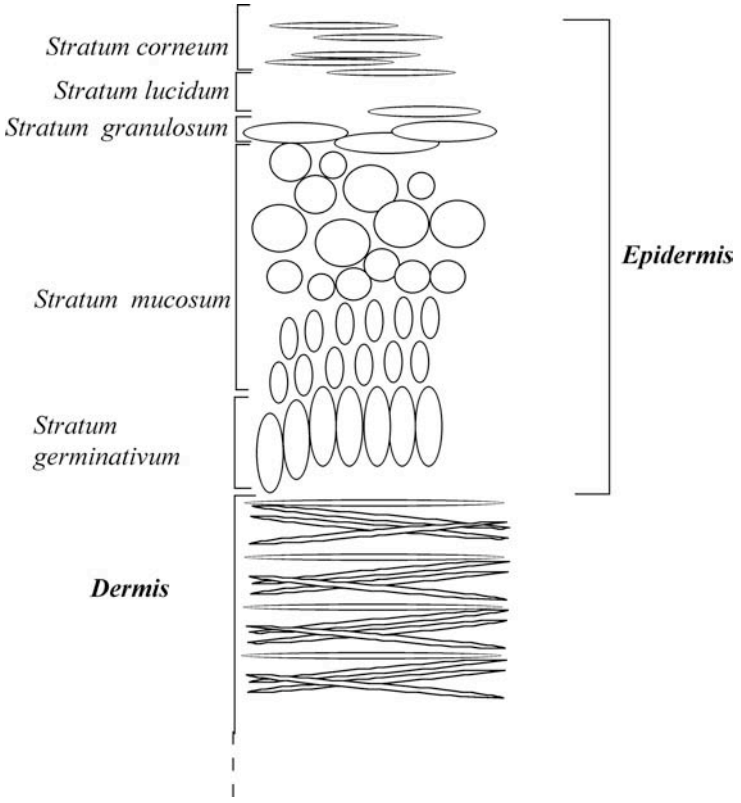

Fig. 7.1 - Skin can be modeled as a spheroids of different sizes, a Monte Carlo program will be written to describe the effect on polarization of these layers.

The Monte Carlo program will be also modified to deal with birefringent collagen layers. The collagen bundles will be modeled as cylinders with a set birefringence and a random fast axis orientation.

These Monte Carlo models could be easily modified to handle polarized florescence, two groups are currently working on this issue.

7.4. Heuristic model of polarized light transport in mono disperse solutions of micro spheres and scattering media

Monte Carlo programs for polarized light transport in scattering media are useful, but so far are limited to Mie and Rayleigh scatterers. A heuristic model to predict, and with some accuracy, the amount of polarized light through a slab or reflected from a semi-infinite medium would be of great interest to the biomedical optics community.

Jacques *et al.* have suggested a model based on a diffusivity parameter χ. In this chapter two models were shown to work quite well on a range of mono disperse solutions, and one model was successfully tested on biological tissues. In order to understand how polarized light distributes when traveling through a slab, several Monte Carlo simulations were conducted (Figure4.11 to 4.20). These simulations show a very structured distribution of polarization for micro sphere solutions strongly depending on the particle size. The mechanism of depolarization due to scattering from small and large spheres was also studied. Small particles tend to transform incident linearly polarized scattered light into different linear states of polarization. Large particles tend transform the incident light in both linear and elliptical states of polarization. The second heuristic model is very promising but needs further testing.

Future work should be done in adapting the heuristic models for reflectance experiments for both sphere solutions and biological media.

7.5. Depth of penetration of polarized light in biological tissues and macro retardation tests

In chapter 5, a series of experiments was conducted to establish the depolarization depth for three biological tissues at different wavelengths. A small wavelength dependence was shown with longer wavelengths propagating deeper that shorter ones as expected. One set of experiments in the visible range (550 nm) showed that light was completely depolarized after ~120 μm for skin, ~200 μm form muscle, and ~250μm for liver. Polarized light imaging uses the shallow depth of penetration to focus the image on the very superficial layer of the skin, one could argue that the same effect could be obtained with short wavelengths (λ< 500 nm), the advantage of polarized light imaging relies on the enhancement of contrast. In Figure 5.6 a comparison of contrast efficiency for polarized and unplarized imaging was shown. The contrast in polarized light imaging was twice as big as the one for unpolarized light.

Experiments were also conducted with a polarimetry system to reconstruct the Mueller matrix of a biological tissue slab. One goal of this experiment was to come up

with a polarimetric signature for biological tissue to include in Monte Carlo programs. Three biological media: skin, liver, and muscle were studied and the tissue were modeled as a simple retarder. This modeled showed not to be a valid one since a general attenuation was obtained when passing through tissue of increasing thickness. For a retarder model a more periodical behavior was expected.

Experiments were also conducted on biological tissues with a polarized microscope. These experiments displayed how certain tissues exhibit micro-domain of iso-birefringence on the order of 10~100mm in size.

7.6. Asymmetric microscope to study precancerous nuclei enlargement

In the last few years several groups began investigating polarized light backscattered from cells and sub-cellular particulates in a precancerous state as an early marker for certain cancers. Previous work herein discussed how variable polarized light transmission and reflectance from bulk tissue can be. In this new experiment a microscope was used to look at scattering from diluted solutions of micro spheres to prove that one could predict the particle sizes once the particle index of refraction and incident wavelength was well known. An asymmetric polarized illumination was used to recreate the single particle phase function. With this technique it was shown that particle size could be predicted restricting the numerical aperture of the collecting lens. Some early testing on muscle cells showed a broader scattering from cytoplasm than from a nucleus. This differentiation could enhance the discrimination between the nucleus and surrounding media and ultimately in the sizing of nuclei.

Appendix A
Hand-held prototype control software

A.1 Apple license agreement

This code was developed starting from munggrab.c from apple, that software comes with the following disclaimer:


```
The Apple Software is provided by Apple on an "AS IS" basis.  APPLE MAKES NO
    WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED
    WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR
    PURPOSE, REGARDING THE APPLE SOFTWARE OR ITS USE AND OPERATION ALONE OR IN
    COMBINATION WITH YOUR PRODUCTS.

    IN NO EVENT SHALL APPLE BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR
    CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE
    GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
    ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION
    OF THE APPLE SOFTWARE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT, TORT
    (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF APPLE HAS BEEN
    ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```


Comments are in *italic*

A.1.1 Source code

File:iPOL.c
Author: Jessica C. Ramella-Roman
Description: this program acquires 2 video channels. Both channels have to be active (cameras plugged in) for the program to work. R1 refers to PER, R0 refers to PAR
POL is calculated as (R0-R1)/(R0+R1)
NOTE:
this sample uses carbon accessesors and will not
build if you have not specified a carbon target
build for carbon

```
#define TARGET_API_MAC_CARBON 1

#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <StandardFile.h>
#include <Navigation.h>
#include <QuickTimeComponents.h>
```

```
#if __APPLE_CC__
#include <Carbon/Carbon.h>
#include <QuickTime/QuickTime.h>
#else
#include <ConditionalMacros.h>
#include <Carbon.h>
#include <QuickTimeComponents.h>
#endif

#define BailErr(x) {err = x; if(err != noErr) goto bail;}
#define kFileTypePICT 'TIFF'
#define kFileCreator 'prvw'

Rect tempRect2;
typedef struct {
WindowRef           pWindow;            // window
Rect boundsRect;    // bounds rect
GWorldPtr           pGWorld;            // offscreen
SeqGrabComponent            seqGrab;  // sequence grabber
ImageSequence               decomSeq;
ImageSequence               drawSeq; // unique identifier for our draw sequence
long        drawSize;
TimeValue  lastTime;
TimeScale  timeScale;
long        frameCount;
}
MungDataRecord, *MungDataPtr;

globals
static BitMap gScreenbits;
static MungDataPtr gMungData0 = NULL;
static MungDataPtr gMungData1 = NULL;
static Boolean gDone = false,
            gIsCollapsed = false,
            gIsGrabbing = false;

UInt32 mask = 0xFFFFFFFF;

RectwindowRect = {0, 0, 400, 400};
PixMapHandle       hPixMap,hPixMap0;
int darkFlag;
int filen,bar;
const unsigned char*            defaultName = "\pImage";
int *darkR0;
```

```
int *darkG0;
int *darkB0;

int *darkR1;
int *darkG1;
int *darkB1;
MatrixRecord scaleMatrix;

int *averageR0;
int *averageG0;
int *averageB0;

int *averageR1;
int *averageG1;
int *averageB1;

int count,ncount;
int fname,writeflag;
SeqGrabComponent            seqGrab0,seqGrab1;
SGChannelsgchanVideo0,sgchanVideo1;

WindowRefpMainWindow0 = NULL;
WindowRefpMainWindow1 = NULL;
Rect        portRect0,portRect1;
GraphicsExportComponent theExportComponent =NULL;
StringPtr prompt;
const unsigned char*            fileName = "\pG4laptop";      // Generic filename
NavDialogOptionsdialogOptions;
OSType   fileTypeToSave = 'PICT';
OSType   creatorType = 'ogle';
NavReplyRecordreply;
AEKeyword theKeyword;
FSSpecdocumentFSSpec;
const unsigned char *MyName;
char *status2;
int dx;
int dy;
double angle;
void Initialize(void);

OSErr InitializeMungData(Rect inBounds, WindowRef inWindow);

OSErr MakeAWindow(WindowRef *outWindow);
SeqGrabComponent MakeSequenceGrabber(WindowRef pWindow);
```

```
OSErr MakeSequenceGrabChannel(SeqGrabComponent seqGrab, SGChannel
*sgchanVideo, Rect const *rect);

void DoUpdate(void);
void saveToPICTFile(WindowPtr theWindow,PixMapHandle hPixMap);
OSErr MakeImageSequenceForGWorld(GWorldPtr pGWorld, GWorldPtr pDest, long
*imageSize, ImageSequence *seq);

pascal OSErr MungGrabDataProc(SGChannel c, Ptr p, long len, long *offset, long
chRefCon, TimeValue time, short writeType, long refCon);

OSErr InitializeMungData0(Rect inBounds, WindowRef inWindow);

OSErr InitializeMungData1(Rect inBounds, WindowRef inWindow);

pascal OSErr MungGrabDataProc0(SGChannel c, Ptr p, long len, long *offset, long
chRefCon, TimeValue time, short writeType, long refCon);

pascal OSErr MungGrabDataProc1(SGChannel c, Ptr p, long len, long *offset, long
chRefCon, TimeValue time, short writeType, long refCon);

OSErr MakeAWindow0(WindowRef *outWindow);

OSErr MakeAWindow1(WindowRef *outWindow);

void QTFrame_HandleKeyPress (EventRecord *theEvent);

void SaveImageToDisk(void);

void writePictToFile(FSSpec *fspec, PicHandle picHandle);

OSErr err = noErr;

Initialize for Carbon & QuickTime

void Initialize(void)
{
        InitCursor();
        EnterMovies();
        GetQDGlobalsScreenBits(&gScreenbits);
}
```

```
static unsigned char * c_to_p(char *cstring)
{
        unsigned char *pstring;
        int len,i;
        len=strlen(cstring);
        if (len>254) return NULL;
        pstring=(unsigned char *)malloc(strlen(cstring)+2);
        pstring[0]= (unsigned char) len;
        for (i=0; i<=len; i++)
        pstring[i+1] = cstring[i];
        return pstring;
}

staticchar * p_to_c(unsigned char pstring[])
{
         char *cstring;
        int len,i;
        len=pstring[0];
        cstring=(char *)malloc(len+1);
        for (i=0; i<=len; i++)
cstring[i] = pstring[i+1];
        cstring[len]='\0';
        return cstring;
}
```

SavetoPICT with WritePictToFile Saves images to file in PICT format

```
void saveToPICTFile(WindowRef theWindow, PixMapHandle hPixMap)
{
        PicHandle           picHandle;
        OSErranErr = noErr;
        OSTypefileTypeToSave = 'PICT';
        OSTypecreatorType = 'ogle';// PictureViewer
        Rect tempRect1;
        if (theWindow == NULL || !IsValidWindowPtr(theWindow))
return;
        LockPixels(hPixMap);
        SetPortWindowPort(theWindow);
        GetPortBounds(GetWindowPort(theWindow), &tempRect1);
        picHandle = OpenPicture(&tempRect1);
        CopyBits((BitMap*) *hPixMap,
        GetPortBitMapForCopyBits(GetWindowPort(theWindow)),
                &(*hPixMap)->bounds,
                &tempRect1,srcCopy, 0);
```

```
            ClosePicture();
            writePictToFile(&documentFSSpec, picHandle);
            reply.translationNeeded = false;
            anErr = NavCompleteSave(&reply, kNavTranslateInPlace);
            NavDisposeReply(&reply);
            KillPicture(picHandle);
            UnlockPixels(hPixMap);
}

void writePictToFile(FSSpec *fspec, PicHandle picHandle)
{
            OSErr anErr = noErr;
            longinOutCount;
            shortrefNum;
            OSType fileTypeToSave = 'PICT';
            OSTypecreatorType = 'ogle'; // PictureViewer
            intscount;
            unsigned char         header[512];

Pict files have to have 512 bytes of "zero" data at the front.

            for (scount = 0; scount < 512; scount++)
            header[scount] = 0x00;

            anErr = FSpCreate(fspec, creatorType, fileTypeToSave, smSystemScript);
            if (anErr == dupFNErr) {
                        anErr = FSpDelete(fspec);      //delete file if already exists
                        anErr = FSpCreate(fspec, creatorType, fileTypeToSave,
                        smSystemScript);
            }

write the file
FSpOpenDF(fspec, fsRdWrPerm, &refNum );
inOutCount = 512;

anErr= FSWrite(refNum, &inOutCount, header); // write the header

if (anErr == noErr) {
            inOutCount = GetHandleSize((Handle)picHandle);
            anErr = FSWrite(refNum,&inOutCount,*picHandle);
            }
FSClose( refNum );
}
```

```
Initialize video channel 0

OSErr InitializeMungData0(Rect inBounds, WindowRef inWindow)
{
        CGrafPtr theOldPort;
        GDHandle theOldDevice;
        OSErr err = noErr;

allocate memory for the data
        gMungData0 = (MungDataPtr)NewPtrClear(sizeof(MungDataRecord));
        if (MemError() || NULL == gMungData0 ) return NULL;
create a GWorld
        err = QTNewGWorld(&(gMungData0->pGWorld),          //returned GWorld
                k32ARGBPixelFormat,         // pixel format
                &inBounds,// bounds
                0,          // color table
                NULL,    // GDHandle
                0);         // flags
        BailErr(err);

lock the pixmap and make sure it's locked because
 you can't decompress into an unlocked pixmap
        if(!LockPixels(GetGWorldPixMap(gMungData0->pGWorld)))
        goto bail;
        GetGWorld(&theOldPort, &theOldDevice);
        SetGWorld(gMungData0->pGWorld, NULL);
        BackColor(blackColor);
        ForeColor(whiteColor);
        EraseRect(&inBounds);
        SetGWorld(theOldPort, theOldDevice);
        gMungData0->boundsRect = inBounds;
        gMungData0->pWindow = inWindow;
        bail:
        return err;
}
QTFrame_HandleKeyPress
Handle keyboard keys and mouse clicks.
This is modelled on Inside Macintosh: Macintosh Toolbox Essentials, p. 3-78.

void QTFrame_HandleKeyPress (EventRecord *theEvent)
{
        charmyKey;
        int offset = 1;
        double rot=1;
```

```
myKey = (theEvent->message) & charCodeMask;
if (theEvent->modifiers & optionKey)
offset =10;
switch (myKey) {
case '\37' : dy -= offset;
break;
case '\36' : dy += offset;
break;
case '\35' : dx -= offset;
break;
case '\34' : dx += offset;
break;
case '1' : ncount=1;
SetWTitle(pMainWindow1, "\pPol no average");
count=1;
break;
case '2' : ncount=2;
SetWTitle(pMainWindow1, "\pPol average of 2");
count=1;
break;
case '3' : ncount=3;
SetWTitle(pMainWindow1, "\pPol average of 3");
count=1;
break;
case '4' : ncount=4;
SetWTitle(pMainWindow1, "\pPol average of 4");
count=1;
break;
case '5' : ncount=5;
SetWTitle(pMainWindow1, "\pPol average of 5");
count=1;
break;
case '0' : ncount=20;
SetWTitle(pMainWindow1, "\pPol average of 20");
count=1;
break;
case '+' :bar+=20;
break;
case '-' :bar-=20;
break;
case 'd' : darkFlag=1;
break;
case 'r' : mask = 0x00FF0000;
break;
```

```
            case 'g' : mask = 0x0000FF00;
            break;
            case 'b' : mask = 0x000000FF;
            break;
            case 'n' :mask = 0xFFFFFFFF;
            break;
            case 's' : {
            writeflag=1;
            break;
            }
            default : ;
   }
}
```

Initialize video channel 1

```
OSErr InitializeMungData1(Rect inBounds, WindowRef inWindow)
{
            CGrafPtr theOldPort;
            GDHandle theOldDevice;
            OSErr err = noErr;
```

allocate memory for the data

```
            gMungData1 = (MungDataPtr)NewPtrClear(sizeof(MungDataRecord));
            if (MemError() || NULL == gMungData1) return NULL;
```

create a GWorld

```
            err = QTNewGWorld(&(gMungData1->pGWorld),// returned GWorld
                        k32ARGBPixelFormat,// pixel format
                        &inBounds,// bounds
                        0,// color table
                        NULL,// GDHandle
                        0);          // flags
            BailErr(err);
```

lock the pixmap and make sure it's locked because we can't decompress into an unlocked pixmap

```
if(!LockPixels(GetGWorldPixMap(gMungData1->pGWorld)))
            goto bail;

GetGWorld(&theOldPort, &theOldDevice);
SetGWorld(gMungData1->pGWorld, NULL);
BackColor(blackColor);
ForeColor(whiteColor);
```

```
EraseRect(&inBounds);
SetGWorld(theOldPort, theOldDevice);
gMungData1->boundsRect = inBounds;
gMungData1->pWindow = inWindow;
bail:
return err;
}

 MakeImageSequenceForGWorld

OSErr MakeImageSequenceForGWorld(GWorldPtr pGWorld,
            GWorldPtr pDest,
            long *imageSize,
            ImageSequence *seq)
{
ImageDescriptionHandle desc = NULL;
PixMapHandle hPixMap = GetGWorldPixMap(pGWorld);
Rect bounds;
OSErr err = noErr;
GetPixBounds(hPixMap, &bounds);
*seq = NULL;

 returns an image description for the GWorlds PixMap
 on entry the imageDesc is NULL, on return it is correctly filled out
 you are responsible for disposing it

err = MakeImageDescriptionForPixMap(hPixMap, &desc);
BailErr(err);

*imageSize = (GetPixRowBytes(hPixMap) * (*desc)->height);

 begin the process of decompressing a sequence of frames
 the destination is the onscreen window

err = DecompressSequenceBegin(seq,
 pointer to field to receive unique ID for sequence
            desc,       // handle to image description structure
            pDest,      // port for the DESTINATION image
            NULL,    // graphics device handle, if port is set, set to NULL
            &bounds,// source rectangle defining the portion of the image to decompress
            NULL,    // transformation matrix
            ditherCopy,// transfer mode specifier
            (RgnHandle)NULL,            // clipping region in dest. coordinate system to
 use as a mask
```

```
            0,          // flags
            codecNormalQuality,         // accuracy in decompression
            anyCodec);// compressor identifier or special identifiers ie.bestSpeedCodec
bail:
if (desc)
            DisposeHandle((Handle)desc);

return err;
}
```

sequence grabber data procedure - this is where the work is done
MungGrabDataProc - the sequence grabber calls the data function whenever
any of the grabber's channels write digitized data to the destination movie file.

NOTE: We really mean any, if you have an audio and video channel then the DataProc will be called for either channel whenever data has been captured. Be sure to check which
channel is being passed in. In this example we never create an audio channel so we know we're always dealing with video.
This data function does two things, it first decompresses captured video
data into an offscreen GWorld, draws some status information onto the frame then
transfers the frame to an onscreen window.

For more information refer to Inside Macintosh: QuickTime Components, page 5-120
c - the channel component that is writing the digitized data.
p - a pointer to the digitized data.
len - the number of bytes of digitized data.
offset - a pointer to a field that may specify where you are to write the digitized data,
and that is to receive a value indicating where you wrote the data.
chRefCon - per channel reference constant specified using SGSetChannelRefCon.
time - the starting time of the data, in the channel's time scale.
writeType - the type of write operation being performed.
seqGrabWriteAppend - Append new data.
seqGrabWriteReserve - Do not write data. Instead, reserve space for the amount of data specified in the len parameter.
seqGrabWriteFill - Write data into the location specified by offset. Used to fill the space previously reserved with seqGrabWriteReserve. The Sequence Grabber may call the DataProc several times to fill a single reserved location.
refCon - the reference constant you specified when you assigned your data function to the sequence grabber.
*/

```
pascal OSErr MungGrabDataProc0(SGChannel c, Ptr p, long len, long *offset, long chRefCon, TimeValue time, short writeType, long refCon)
{
```

```
#pragma unused(offset,chRefCon,writeType,refCon)

CGrafPtr theSavedPort;
GDHandletheSavedDevice;
CodecFlags            ignore;
floatfps = 0, averagefps         = 0;
charstatus[64];
Str255theString;
PixMapHandle       PixMap;
ComponentResult err = noErr;
static MungDataPtr gMungData = NULL;
gMungData=gMungData0;

 reset frame and time counters after a stop/start
if (gMungData->lastTime > time) {
            gMungData->lastTime = 0;
            gMungData->frameCount = 0;
}

gMungData->frameCount++;
if (gMungData->timeScale == 0) {
// first time here so set the time scale
err = SGGetChannelTimeScale(c, &gMungData->timeScale);
            BailErr(err);
}
if (gMungData->pGWorld) {
if (gMungData->decomSeq == 0) {
 Set up getting grabbed data into the GWorld
Rect sourceRect = { 0, 0 };
MatrixRecord scaleMatrix;
ImageDescriptionHandle imageDesc = (ImageDescriptionHandle)NewHandle(0);

 retrieve a channel's current sample description, the channel returns a sample
description that is appropriate to the type of data being captured
err = SGGetChannelSampleDescription(c, (Handle)imageDesc);
BailErr(err);
 make a scaling matrix for the sequence

 sourceRect.right = (**imageDesc).width;
 sourceRect.bottom = (**imageDesc).height;
 RectMatrix(&scaleMatrix, &sourceRect, &gMungData->boundsRect);
 angle     =           -0.89;
RotateMatrix(&scaleMatrix, X2Fix(angle),X2Fix(218),X2Fix(198) );
SkewMatrix(&scaleMatrix, X2Fix(0),X2Fix(-.01),X2Fix(0),X2Fix(0) );
```

```
err = DecompressSequenceBegin(&gMungData->decomSeq,
            imageDesc,
            gMungData->pGWorld,
            NULL,
            NULL,
            &scaleMatrix,
            srcCopy,
            (RgnHandle)NULL,
            NULL,
            codecNormalQuality,
            bestSpeedCodec);.
BailErr(err);
DisposeHandle((Handle)imageDesc);

err = MakeImageSequenceForGWorld(gMungData->pGWorld,
            GetWindowPort(gMungData->pWindow),
            &gMungData->drawSize,
            &gMungData->drawSeq);
BailErr(err);
}

 decompress a frame into the GWorld - can queue a frame for async decompression when
passed in a completion proc
err = DecompressSequenceFrameS(gMungData->decomSeq,
            p,
            len,
            0,
            &ignore,
            NULL);

if (err) {
            TextSize(10);
            TextMode(srcXor);
            MoveTo(gMungData->boundsRect.left + 10, gMungData->boundsRect.top +
            80);
            sprintf(status,"DecompressSequenceFrameS gave error %ld (%lx)",err,err);
            CopyCStringToPascal(status, theString);
            DrawString(theString);
            err = noErr;
            } else {
             write status information onto the frame
            GetGWorld(&theSavedPort, &theSavedDevice);
            SetGWorld(gMungData->pGWorld, NULL);
```

```
            TextSize(12);
            TextMode(srcCopy);
            MoveTo(gMungData->boundsRect.left +10, gMungData->boundsRect.bottom
            - 14);
            fps = (float)gMungData->timeScale / (float)(time - gMungData->lastTime);
            averagefps = ((float)gMungData->frameCount * (float)gMungData-
            >timeScale) / (float)time;
            hPixMap = GetGWorldPixMap(gMungData->pGWorld);
 draw the frame to the destination, in this case the onscreen window
            err = DecompressSequenceFrameS(gMungData->drawSeq, // sequence ID
            GetPixBaseAddr(GetGWorldPixMap(gMungData->pGWorld)),
            gMungData->drawSize,// size of the buffer
            0,          // in flags
            &ignore,// out flags
            NULL);
 draw the frame to the destination, in this case the onscreen window
            }
}

bail:
gMungData->lastTime = time;
            return err;
}

pascal OSErr MungGrabDataProc1(SGChannel c, Ptr p, long len, long *offset, long
            chRefCon, TimeValue time, short writeType, long refCon)
{
#pragma unused(offset,chRefCon,writeType,refCon)

CodecFlagsignore;
float       fps = 0,
            averagefps = 0;
char        status[64];
            Str255     theString;
long        theRowBytes,theRowBytes0,height,width;
            PtrpPixels,pPixels0;
            ComponentResult err = noErr;
            Rect       bounds;
            MungDataPtr         gMungData = gMungData1;
            intx,y;
            UInt32     R0,G0,B0,R1,G1,B1;
            UInt32     background=0xFF0000FF;
            OSType     fileTypeToSave = 'TIFF';
            OSType     creatorType = 'ogle';// PictureViewer
```

```
 reset frame and time counters after a stop/start
if (gMungData->lastTime > time) {
            gMungData->lastTime = 0;
            gMungData->frameCount = 0;
}
gMungData->frameCount++;
if (gMungData->timeScale == 0) {
 first time here so set the time scale
            err = SGGetChannelTimeScale(c, &gMungData->timeScale);
            BailErr(err);
}
if (gMungData->pGWorld) {
if (gMungData->decomSeq == 0) {
/ Set up getting grabbed data into the GWorld
            Rect sourceRect = { 0, 0 };
            MatrixRecord scaleMatrix;
            ImageDescriptionHandle imageDesc =
(ImageDescriptionHandle)NewHandle(0);

 retrieve a channel's current sample description, the channel returns a sample
description that is
 appropriate to the type of data being captured
err = SGGetChannelSampleDescription(c, (Handle)imageDesc);
BailErr(err);


 make a scaling matrix for the sequence
sourceRect.right = (**imageDesc).width;
sourceRect.bottom = (**imageDesc).height;
RectMatrix(&scaleMatrix, &sourceRect, &gMungData->boundsRect);

 begin the process of decompressing a sequence of frames
err = DecompressSequenceBegin(&gMungData->decomSeq,
            imageDesc,                      // handle to image description structure
            gMungData->pGWorld,     // port for the DESTINATION image
            NULL,               // graphics device handle, if port is set, set NULL
            NULL,               // source rectangle defining the portion of the image to
            &scaleMatrix, // transformation matrix
            srcCopy,// transfer mode specifier
            (RgnHandle)NULL,
            NULL,
            codecNormalQuality,          // accuracy in decompression
            bestSpeedCodec);
```

```
BailErr(err);
DisposeHandle((Handle)imageDesc);

// Set up getting grabbed data into the Window
// create the image sequence for the offscreen
err = MakeImageSequenceForGWorld(gMungData->pGWorld,
            GetWindowPort(gMungData->pWindow),
            &gMungData->drawSize,
            &gMungData->drawSeq);
BailErr(err);
}

// decompress a frame into the GWorld - can queue a frame for async decompression
when passed in a completion proc
err = DecompressSequenceFrameS(gMungData->decomSeq, len,0,&ignore,NULL);

if (err) {
            TextSize(10);
            TextMode(srcXor);
            MoveTo(gMungData->boundsRect.left + 10, gMungData->boundsRect.top +
            80);
            sprintf(status,"DecompressSequenceFrameS gave error %ld (%lx)",err,err);
            CopyCStringToPascal(status, theString);
            DrawString(theString);
            err = noErr;
} else {
            hPixMap0 = GetGWorldPixMap(gMungData0->pGWorld);
            theRowBytes0 = QTGetPixMapHandleRowBytes(hPixMap0);
            pPixels0 = GetPixBaseAddr(hPixMap0);
            hPixMap= GetGWorldPixMap(gMungData->pGWorld);
            theRowBytes= QTGetPixMapHandleRowBytes(hPixMap);
            pPixels= GetPixBaseAddr(hPixMap);
            GetPixBounds(hPixMap,&bounds);
            height = (bounds.bottom - bounds.top);
            width = (bounds.right - bounds.left);

for (y = 0; y < height; ++y) {UInt32 *p,*p0;p = (UInt32*)(pPixels + theRowBytes *
(long)y);
p0= (UInt32*)(pPixels0 + theRowBytes0 * (long)(y+dy))+dx;
for (x = 0; x < width; x++, p++, p0++) {
            R0 = (*p0 & 0x00FF0000) >> 16;
            G0 = (*p0 & 0x0000FF00) >> 8;
            B0 = (*p0 & 0x000000FF) >> 0;
            R1 = (*p & 0x00FF0000) >> 16;
```

```
            G1 = (*p & 0x0000FF00) >> 8;
            B1 = (*p & 0x000000FF) >> 0;
            averageR0[x+width*y] += R0;
            averageG0[x+width*y] += G0;
            averageB0[x+width*y] += B0;
            averageR1[x+width*y] += R1;
            averageG1[x+width*y] += G1;
            averageB1[x+width*y] += B1;
            }
}

// If the averaging is acheived
 if (count ==ncount){
for (y = 0; y < height; ++y) {
            UInt32 *p,*p0,R,G,B,pg,pr,pb;
            p = (UInt32*)(pPixels + theRowBytes * (long)y);
            p0= (UInt32*)(pPixels0 + theRowBytes0 * (long)(y+dy))+dx;
            for (x = 0; x < width; x++, p++, p0++) {
                        if (x+dx<0 || x+dx>width||y+dy<0 || y+dy>height) {
                        *p=background;
                        continue;
                        }

if (writeflag==1){
R=(UInt32)fabs( (float)(averageR1[x+width*y])/(float)(ncount) );
G=(UInt32)fabs( (float)(averageG1[x+width*y])/(float)(ncount) );
B=(UInt32)fabs( (float)(averageB1[x+width*y])/(float)(ncount) );
}else{
R=(UInt32)bar*(((float)(averageR0[x+width*y])-
(float)(averageR1[x+width*y]))/((float)(averageR0[x+width*y])+(float)(averageR1[x+wi
dth*y])));
if (R<0)
R=0.0001;
G=(UInt32)bar*(((float)(averageG0[x+width*y])-
(float)(averageG1[x+width*y]))/((float)(averageG0[x+width*y])+(float)(averageG1[x+w
idth*y])));
if (G<0)
G=0.0001;
B=(UInt32)bar*(((float)(averageB0[x+width*y])-
(float)(averageB1[x+width*y]))/((float)(averageB0[x+width*y])+(float)(averageB1[x+wi
dth*y])));
if (B<0)
B=0.0001;
}
```

```
pr = R<<16;
pg = G<<8;
pb = B;
*p=pr+pg+pb;
*p&=mask;

averageR0[x+width*y]=0;
averageG0[x+width*y]=0;
averageB0[x+width*y]=0;

averageR1[x+width*y]=0;
averageG1[x+width*y]=0;
averageB1[x+width*y]=0;

}
}
}
if (count ==ncount){
Rect tempRect1;
CopyBits((BitMap *)
*hPixMap,GetPortBitMapForCopyBits(GetWindowPort(pMainWindow1)),
&bounds,GetPortBounds(GetWindowPort(pMainWindow1),&tempRect1),notSrcCopy,
0);
count=0;

if (writeflag==1){
char status[64];
fname+=1;
sprintf(status,"%s%d",status2,fname);
CopyCStringToPascal(status, documentFSSpec.name);
saveToPICTFile(pMainWindow1,hPixMap);

//Save also Par

sprintf(status,"%s%dpar",status2,fname);
CopyCStringToPascal(status, documentFSSpec.name);
saveToPICTFile(pMainWindow0,hPixMap0);
writeflag=0;
            }
}
}
}
count+=1;
bail:
```

```
gMungData->lastTime = time;

        return err;
}


/* --------------------
 DoUpdate
*/

void DoUpdate(void)
{
        CodecFlags      ignore;
// draw the last frame captured
        DecompressSequenceFrameS(gMungData0->drawSeq,
         GetPixBaseAddr(GetGWorldPixMap(gMungData0->pGWorld)),
        gMungData0->drawSize,
        0,
        &ignore,
        NULL);
        DecompressSequenceFrameS(gMungData1->drawSeq,
         GetPixBaseAddr(GetGWorldPixMap(gMungData1->pGWorld)),
        gMungData1->drawSize,
        0,
        &ignore,
        NULL);
}
/* --------------------
 MakeSequenceGrabber
*/
SeqGrabComponent MakeSequenceGrabber(WindowRef pWindow)
{
        SeqGrabComponent seqGrab = NULL;
        OSErr   err = noErr;

// open the default sequence grabber
seqGrab = OpenDefaultComponent(SeqGrabComponentType, 0);
if (seqGrab != NULL) {
        err = SGInitialize(seqGrab);
        if (err == noErr)
        err = SGSetGWorld(seqGrab, GetWindowPort(pWindow), NULL );
        if (err == noErr)
        err = SGSetDataRef(seqGrab,
```

```
                0,
                0,
                seqGrabDontMakeMovie);
        }

if (err && (seqGrab != NULL)) { // clean up on failure
        CloseComponent(seqGrab);
seqGrab = NULL;
}
        return seqGrab;
}

// --------------------
// MakeSequenceGrabChannel
//
OSErr MakeSequenceGrabChannel(SeqGrabComponent seqGrab, SGChannel
*sgchanVideo, Rect const *rect)
{
longflags = 0;

OSErr err = noErr;

err = SGNewChannel(seqGrab, VideoMediaType, sgchanVideo);
if (err == noErr) {
        err = SGSetChannelBounds(*sgchanVideo, rect);
        if (err == noErr)
// set usage for new video channel to avoid playthrough
         // note we don't set seqGrabPlayDuringRecord
err = SGSetChannelUsage(*sgchanVideo, flags | seqGrabRecord );

        if (err != noErr) {
        // clean up on failure
        SGDisposeChannel(seqGrab, *sgchanVideo);
        *sgchanVideo = NULL;
        }
}
        return err;
}

// --------------------
// MakeAWindow
//
OSErr MakeAWindow0(WindowRef *outWindow)
{
```

```
RectbestRect;
OSErr      err = noErr;
// Figure out the best monitor for the window
GetBestDeviceRect(NULL, &bestRect);
// put the window in the top left corner of that monitor
OffsetRect(&windowRect, bestRect.left+500 , bestRect.top );
 err = CreateNewWindow(kDocumentWindowClass, kWindowCloseBoxAttribute,
              &windowRect, outWindow);
              BailErr(err);
              SetWTitle(*outWindow, "\pH IMAGE");
// set the port to the new window
SetPortWindowPort(*outWindow);
ShowWindow(*outWindow);
bail:
return err;
}
//
OSErr MakeAWindow1(WindowRef *outWindow)
{
RectbestRect;
OSErr      err = noErr;
// Figure out the best monitor for the window
GetBestDeviceRect(NULL, &bestRect);

// put the window in the top left corner of that monitor
OffsetRect(&windowRect, bestRect.left + 100, bestRect.top + 50);
              err = CreateNewWindow(kDocumentWindowClass,
              kWindowCloseBoxAttribute, &windowRect, outWindow);
              BailErr(err);
              SetWTitle(*outWindow, "\pPOL IMAGE");

// set the port to the new window
SetPortWindowPort(*outWindow);
ShowWindow(*outWindow);
bail:
return err;
}

/*-----------------------------------------------
Main
*/
int main(void)
{
```

```
OSErr     err = noErr;
SGDeviceList list0,list1;
int       k,i;
SGDeviceName myDevice,device;
char      *str;
int       camera;
DescTypeactualType;
SizeactualSize;
bar       = 255;
writeflag = 0;
darkFlag  = 0;

averageR0          =(int *)malloc(400*400*sizeof(int));
averageG0          =(int *)malloc(400*400*sizeof(int));
averageB0          =(int *)malloc(400*400*sizeof(int));

averageR1          =(int *)malloc(400*400*sizeof(int));
averageG1          =(int *)malloc(400*400*sizeof(int));
averageB1          =(int *)malloc(400*400*sizeof(int));

darkR0   =(int *)malloc(400*400*sizeof(int));
darkG0   =(int *)malloc(400*400*sizeof(int));
darkB0   =(int *)malloc(400*400*sizeof(int));

darkR1   =(int *)malloc(400*400*sizeof(int));
darkG1   =(int *)malloc(400*400*sizeof(int));
darkB1   =(int *)malloc(400*400*sizeof(int));

for(i=0;i<(400*400);i++){

averageR0[i]=0;
averageG0[i]=0;
averageB0[i]=0;

averageR1[i]=0;
averageG1[i]=0;
averageB1[i]=0;

darkR0[i]=0;
darkG0[i]=0;
darkB0[i]=0;

darkR1[i]=0;
darkG1[i]=0;
```

```
darkB1[i]=0;

}

            Initialize();
            count=0;
            ncount=1;
            angle=0;
            dx = 0;
            dy = -22;


NavGetDefaultDialogOptions(&dialogOptions);
            dialogOptions.dialogOptionFlags |= kNavSelectDefaultLocation;
            PLstrcpy(dialogOptions.savedFileName, defaultName);

            NavPutFile( nil,
&reply,
&dialogOptions,
nil,
fileTypeToSave,
creatorType,
nil );


            if (err == noErr && reply.validRecord) {
err = AEGetNthPtr(&(reply.selection), 1, typeFSS,
&theKeyword, &actualType,
&documentFSSpec, sizeof(documentFSSpec),
&actualSize );
}
MyName=documentFSSpec.name;
status2=p_to_c((unsigned char *)MyName);

err = MakeAWindow1(&pMainWindow1);
BailErr(err);
err = MakeAWindow0(&pMainWindow0);
BailErr(err);

GetPortBounds(GetWindowPort(pMainWindow0), &portRect0);
GetPortBounds(GetWindowPort(pMainWindow1), &portRect1);

// initialize our data
err = InitializeMungData0(portRect0, pMainWindow0);
```

```
BailErr(err);
err = InitializeMungData1(portRect1, pMainWindow1);
BailErr(err);

// create and initialize the sequence grabber
seqGrab0 = MakeSequenceGrabber(pMainWindow0);
BailErr(NULL == seqGrab0);
seqGrab1= MakeSequenceGrabber(pMainWindow1);
BailErr(NULL == seqGrab1);

err = MakeSequenceGrabChannel(seqGrab0, &sgchanVideo0, &portRect0);
BailErr(err);
err = MakeSequenceGrabChannel(seqGrab1, &sgchanVideo1, &portRect1);
BailErr(err);

SGGetChannelDeviceList(sgchanVideo0,0,&list0);
SGGetChannelDeviceList(sgchanVideo1,0,&list1);

for (k=0; k<(**list0).count; k++) {
        myDevice = (**list0).entry[k];
        str=p_to_c(myDevice.name);
        free(str);
}

camera=2; //This is specific of this setup,
device=(**list0).entry[camera];
SGSetChannelDevice(sgchanVideo0,device.name);//Select camera
err = SGSetDataProc(seqGrab0, NewSGDataUPP(MungGrabDataProc0), 0);
BailErr(err);
// lights...camera...
err = SGPrepare(seqGrab0, false, true);
BailErr(err);


camera=1;
device=(**list1).entry[camera];
SGSetChannelDevice(sgchanVideo1,device.name);//Select camera
err = SGSetDataProc(seqGrab1, NewSGDataUPP(MungGrabDataProc1), 0);
BailErr(err);
err = SGPrepare(seqGrab1, false, true);
BailErr(err);

SGSettingsDialog( seqGrab0,sgchanVideo0, 0, NULL,DoTheRightThing, NULL, 0);
SGSettingsDialog( seqGrab1,sgchanVideo1, 0, NULL,DoTheRightThing, NULL, 0);
```

```
err = SGStartRecord(seqGrab0);
BailErr(err);

err = SGStartRecord(seqGrab1);
BailErr(err);
gIsGrabbing = true;

while (!gDone) {
EventRecord          theEvent;
WindowRef theWindow;
GetNextEvent(everyEvent, &theEvent);
// display polarization each time through loop
if (IsWindowCollapsed(pMainWindow1)) {
            // checking this here avoids codecNothingToBlitErr later
            SGStop(seqGrab0);
            gIsGrabbing = false;
            gIsCollapsed = true;
}

switch (theEvent.what) {
case nullEvent:
            // give the sequence grabber time to do it's thing
            if (gIsGrabbing) {
err = SGIdle(seqGrab0);
err = SGIdle(seqGrab1);

if (err) {
            char errMsg[32];

            // if there is an error, display the result in the window title
            // if it's a cDepthErr we don't pause; the sequence grabber
            // would return cDepthErr if the window was moved or depth changed on
            // QT 4.1.2, it does it less on QT 5 because Kevin made it smarter
            // all other errors cause a pause - errors set in the DataProc show up
            // here as well as others generated by the vDig - different vDigs can
            // generate different errors in different situations
            if (err == cDepthErr) {
sprintf(errMsg, "cDepthErr ", err);
c2pstrcpy((unsigned char *)&errMsg, errMsg);
SetWTitle(pMainWindow0, (unsigned char *)errMsg);
SetWTitle(pMainWindow1, (unsigned char *)errMsg);

SGStop(seqGrab0);
SGStop(seqGrab1);
```

```
SGStartRecord(seqGrab0);
SGStartRecord(seqGrab1);
break;
            } else {
KeyMap  theKeys;
#define ISESCKEYDOWN() ((theKeys[1] & 0x00001000) == 0x00001000)

SGStop(seqGrab0);
SGStop(seqGrab1);

sprintf(errMsg, "Stopped, esc to continue %d", err);
c2pstrcpy((unsigned char *)&errMsg, errMsg);
SetWTitle(pMainWindow0, (unsigned char *)errMsg);
SetWTitle(pMainWindow1, (unsigned char *)errMsg);

// wait for esc
do {
            GetKeys(theKeys);
} while(!ISESCKEYDOWN());
SetWTitle(pMainWindow0, "\pCamera1");
SGStartRecord(seqGrab0);

SetWTitle(pMainWindow1, "\pCamera2");
SGStartRecord(seqGrab1);
            }
}
            }
            break;

case updateEvt:
            theWindow = (WindowRef)theEvent.message;
            if (theWindow == pMainWindow0 || theWindow == pMainWindow1) {
            if (gIsGrabbing) {
// inform the sequence grabber of the update
            RgnHandle theUpdateRgn = NewRgn();
            GetWindowRegion(theWindow, kWindowUpdateRgn, theUpdateRgn);
            SGUpdate(seqGrab0, theUpdateRgn);
            DisposeRgn(theUpdateRgn);
} else {
            if (!IsWindowCollapsed(pMainWindow0) && gIsCollapsed) {
            // window was just un-collapsed, start grabbing again
            SGStartRecord(seqGrab0);
            gIsGrabbing = true;
```

```
            gIsCollapsed = false;
            } else {
// update the still image
DoUpdate();
            }
            }

            // swallow the update event
BeginUpdate(theWindow);
EndUpdate(theWindow);
}
break;

case keyDown:
case autoKey:
QTFrame_HandleKeyPress(&theEvent);
break;
            case mouseDown:
            short nPart;
            nPart = FindWindow(theEvent.where, &theWindow);

            if (pMainWindow1 == theWindow) {

writeflag=1;

            }

            if (pMainWindow0 != theWindow)
break;
switch (nPart) {
case inGoAway:
gDone = TrackGoAway(theWindow, theEvent.where);
break;
case inDrag:
            ICMAlignmentProcRecord apr;
            SGGetAlignmentProc(seqGrab0, &apr);
DragAlignedWindow(theWindow, theEvent.where, &gScreenbits.bounds, NULL, &apr);
break;
}
break;
case osEvt:
if ((theEvent.message & (suspendResumeMessage << 24)) != 0 ) {
            if ((theEvent.message & resumeFlag) != 0 ) {
if (!gIsGrabbing) {
```

```
// switched in, start grabbin'
            SGStartRecord(seqGrab0);
            SGStartRecord(seqGrab1);
            gIsGrabbing = true;
            }
} else {
if (gIsGrabbing) {
// switched out, stop grabbin'
            SGStop(seqGrab0);
            SGStop(seqGrab1);
            gIsGrabbing = false;
            }
}
}
break;
default:
break;
} // switch
}

bail:
            // clean up
            if (seqGrab0) {
SGStop(seqGrab0);
CloseComponent(seqGrab0);
free (darkR0);
free (averageR0);
free (darkB0);
free (averageB0);
free (darkB0);
free (averageB0);
free (darkR1);
free (averageR1);
free (darkB1);
free (averageB1);
free (darkB1);
free (averageB1);
            }
if (pMainWindow0||pMainWindow1)
            DisposeWindow(pMainWindow0);
            DisposeWindow(pMainWindow1);

            return 0;
}
```

Appendix B
Three Monte Carlo programs

B1. Meridian Planes Monte Carlo

```
/*****************************************************************
 *  Copyright Jessica C. Ramella-Roman, Steve L. Jacques and Scott A. Prahl 2003
 *  Stok1.c -Meridian Planes MC
 *          Main program for Monte Carlo simulation of photon
*           travelling into scattering media keeping track of its status of polarization. Slab
*           geometry.
 *          A report in preparation illustrates use of the program:
 *                      J. Ramella-Roman, S. A. Prahl, S. L. Jacques:
 *          "Three polarized light Monte Carlo programs, a performance comparison,
 *          To be submitted in 2004 Optics Express
 *          This program is free software; you can redistribute it and/or
 *          modify it under the terms of the GNU General Public License
 *          as published by the Free Software Foundation; either version 2
 *          of the License, or (at your option) any later version.
 *          This program is distributed in the hope that it will be useful,
 *          but WITHOUT ANY WARRANTY; without even the implied warranty of
 *          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
*           See the GNU General Public License for more details.
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, write to the Free Software
 *  Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
 *
 ****/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "array.h"
#include "complex.h"
#include "mie.h"
#include "nrutil.h"
#include <time.h>
```

```
#define ALIVE       1
#define DEAD        0
#define    NN          100
#define THRESHOLD   0.01              /* used in roulette */
#define CHANCE      0.1               /* used in roulette */

#define RandomNum (double) RandomGen(1, 0, NULL)
#define SIGN(x) ((x)>=0 ? 1:-1)
#define InitRandomGen (double) RandomGen(0, 1, NULL)

/* Declare Subroutines */
void        rotSphi(double* S, double phi, double* S2);
double      RandomGen(char Type, long Seed, long *Status);
void        multS(double* S, double theta, double* S2);
void        rotateXXYY(double* XX, double* YY,double* ZZ, double phi, double theta,
double* XX2, double* YY2,double* ZZ2);
void        updateU(double* U, double phi, double theta, double* U2);
double  sincos(double *x);

/*************** MAIN ****************************************/
int main() {
            double pi = 3.1415926535897932384;

            /* Mie theory stuff */
            double radius,lambda, A;
            long nangles,i;
            struct complex m;
            struct complex*s1=NULL;
            struct complex*s2=NULL;

            double *mu=NULL;
            double x,qext,qsca,qback,g, rho, vol,dy, dx, hw;
            double nre_p, nim_p, nre_med, nim_med;
            FILE *target;
            double jjj;

            /* E field stuff */
            double      phi, theta,I,I0;
            int         ithedeg;
            double      IT, QT, UT, VT;
            double      IR_1, QR_1, UR_1, VR_1;
            double      **IR, **QR, **UR, **VR
            /* Propagation parameters */
            double      y,z; /* photon position.  x already declared.
```

```
            double     s;          /* step sizes. s = -log(RND)/mus [cm] */
            long       i_photon;   /* current photon */
            long       Nphotons;   /* number of photons in simulation */
            short   photon_status;  /*  = ALIVE=1 or DEAD=0 */

            /* other variables */
            double     mua;        /* absorption coefficient [cm^-1] */
            double     mus;        /* scattering coefficient [cm^-1] */
            double     musp;       /* reduced scattering coefficient [cm^-1] */
            double  albedo;     /* albedo of tissue */

            /* dummy variables */
            int           maxcnt,cnt,totcnt;
            double  rnd;        /* assigned random value 0-1 */
            int     MM;
            double  W,absorb;  /* photon weight */
            double  slabsize;
            int           j,ix,iy;
            double  cos22,sin22,costheta,sini,cosi;

/**** allocate matrices and arrays *******/
            double     *U, *U2;
            double     *S;          /* */
            double     *S0;         /* */
            double     *S2;         /* */
            double     *s11=NULL;
            double     *s12=NULL;
            double     *s33=NULL;
            double     *s43=NULL;
            double     *IQUV;                  /* [I, Q, U, V] Stokes Vector */
            double      start_time,finish_time,temp;

start_time = clock();

MM = NN - 1;

U      = new_darray(3);
U2     = new_darray(3);
S      = new_darray(4);
S0     = new_darray(4);
S2     = new_darray(4);/* dummy S*/
IQUV   = new_darray(4);

IR     = dmatrix(0, MM, 0, MM); /* [0:MM] */
```

```
QR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
UR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
VR    = dmatrix(0, MM, 0, MM); /* [0:MM] */

/**** end  allocate matrices and arrays *******/


/* CHOOSE MIE SCATTERING parameters */
radius              = 2.0/2; /* microns */
lambda              = 0.6328; /* microns */
rho                 = 1.152e-4;/*Dilution 1*/
Nphotons            = 1e4;
mua                 = 0.0; /*μa  */

/* ------------------------*/
nre_p               = 1.59;
nim_p               = 0;
nre_med             = 1.33;
nim_med             = 0.0;
nangles             = 10000;


/* Setup MIE SCATTERING parameters */

mu  = new_darray(nangles);
s1  = new_carray(nangles);
s2  = new_carray(nangles);
s11 = new_darray(nangles);
s12 = new_darray(nangles);
s33 = new_darray(nangles);
s43 = new_darray(nangles);


m.re = nre_p/nre_med;
m.im = 0.0;
x    = 2*pi*radius/(lambda/nre_med);
vol  = 4.0/3*pi*radius*radius*radius;
A    = pi*radius*radius;

for(i=0;i<nangles;i++)
            mu[i] = cos(pi*i/nangles);
            s11=new_darray(nangles);
            s12=new_darray(nangles);
            s33=new_darray(nangles);
```

```
        s43=new_darray(nangles);
        s1=new_carray(nangles);
        s2=new_carray(nangles);

        Mie(x,m,mu,nangles,s1,s2,&qext,&qsca,&qback,&g); /*Call Mie program */

        mus     = qsca*A*rho*1e4; /* Mus is in cm^-1 */
        musp    = mus*(1-g);/* [cm^-1] */
        albedo  = mus/(mus + mua);
        free_darray(mu);

        printf("Polarized Monte Carlo\n dia=%5.5f;\n mus=%5.5f;\n g=%5.5f;\n
rho=%5.5f;\n",radius*2,mus,g,rho);

        slabsize   = 4/mus;/* semiinfinite4/mus; /* cuvette length*/
        printf("Slab size=%5.5f\n",slabsize);

/*Scattering parameters s11 s12 s33 s43*/

for(i=0;i<nangles;++i){
        s11[i] = 0.5*cabbs(s2[i])*cabbs(s2[i]) + 0.5*cabbs(s1[i])*cabbs(s1[i]);
        s12[i] = 0.5*cabbs(s2[i])*cabbs(s2[i]) - 0.5*cabbs(s1[i])*cabbs(s1[i]);
        s33[i] = (cmul(conj(s1[i]),s2[i])).re;
        s43[i] = (cmul(conj(s1[i]),s2[i])).im;
        }

        hw      = 10/mus; /* [cm] , maximum range in x and y for output. */
        dx      = 2.0*hw/NN;
        dy      = 2.0*hw/NN;
        maxcnt  = 50; /*kill photon after maxcnt scattering events*/

/******** MONTE CARLO *******/
        InitRandomGen;
        IT=0;
        QT=0;
        UT=0;
        VT=0;

        IR_1=0;
        QR_1=0;
        UR_1=0;
        VR_1=0;
        temp=0;
```

```
        for (iy=0; iy<NN; iy++){

                for (ix=0; ix<NN; ix++) {
                        IR[iy][ix] = 0.0;
                        QR[iy][ix] = 0.0;
                        UR[iy][ix] = 0.0;
                        VR[iy][ix] = 0.0;
                }
        }

   for (jjj = 1; jjj <= 4; jjj++) {

                if (jjj == 1){

                        S0[0] = 1;
                        S0[1] = 1;
                        S0[2] = 0;
                        S0[3] = 0;
                        printf("launch H\n");}

                if (jjj == 2){
                        S0[0] = 1;
                        S0[1] = -1;
                        S0[2] = 0;
                        S0[3] = 0;
                        printf("launch V\n");}
                if (jjj == 3){
                        S0[0] = 1;
                        S0[1] = 0;
                        S0[2] = 1;
                        S0[3] = 0;
                        printf("launch P\n");}
                if (jjj == 4){
                        S0[0] = 1;
                        S0[1] =    0;
                        S0[2] = 0;
                        S0[3] = 1;
                        printf("launch R\n");}
                        totcnt=0;
/* LAUNCH photon */
                for (i_photon = 1; i_photon <= Nphotons; i_photon++) {

/*pencil beam       */
                x = 0.0;
```

```
                    y = 0.0;
                    z = 0.0;

/* photon direction cosines */
                    U[0] = 0.0;
                    U[1] = 0.0;
                    U[2] = 1.0;
                    for (i=0; i<4; i++) S[i] = S0[i]; /* set incident Stokes vector to S0 */
                    for (i=0; i<4; i++) S2[i] = 0.0; /* set incident Stokes vector to S0 */
                    cnt = 0;
                    photon_status = ALIVE;
                    W       = 1; /* photon weight */

/********* ALIVE cycle ****************/
                    while (photon_status == ALIVE) {
                              cnt+=1;
                              if (cnt>maxcnt){
                                        photon_status = DEAD;
                                        totcnt+=1;
                              }

/**** HOP */
                              rnd = 0; while (rnd == 0) rnd = RandomNum;  /* choose a
step size */

                              s = -log(rnd)/(mus+mua);
                              x += U[0]*s;
                              y += U[1]*s;
                              z += U[2]*s;

/**** ABSORB */

                              absorb = W*(1-albedo);
                              W-= absorb;
if ( z<=0) {
                              /*return to detector reference frame*/
                              phi=atan2(U[1],U[0]);
                              rotSphi(S, phi, S2);

                              IR_1+=S2[0];
                              QR_1+=S2[1];
                              UR_1+=S2[2];
                              VR_1+=S2[3];
```

```
                                if (x >= -hw)
                                        ix = (int)(fabs(x + hw)/dx);
                                if (y >= -hw)
                                        iy = (int)(fabs(y + hw)/dy);
                                if (ix > MM) ix = MM;
                                if (iy > MM) iy = MM;

                                IR[iy][ix] += S2[0];
                                QR[iy][ix] += S2[1];
                                UR[iy][ix] += S2[2];
                                VR[iy][ix] += S2[3];
                                photon_status = DEAD;
        }
        else if ( z>=slabsize) {
                phi=-atan2(U[1],U[0]);
                rotSphi(S, phi, S2);
                IT+=S2[0]*W;
                QT+=S2[1]*W;
                UT+=S2[2]*W;
                VT+=S2[3]*W;
                photon_status = DEAD;
                }/*z>slab size*/
/* SPIN */
/* REJECTION METHOD to choose azimuthal angle phi and deflection angle theta */

                                do{ theta = acos(2*RandomNum-1);
                                phi = RandomNum*2.0*pi;
        I0=s11[0]*S[0]+s12[0]*(S[1]*cos(2*phi)+S[2]*sin(2*phi));
        ithedeg = floor(theta*nangles/pi);
        I=s11[ithedeg]*S[0]+s12[ithedeg]*(S[1]*cos(2*phi)+S[2]*sin(2*phi));
                                }while(RandomNum*I0>=I);

 /*-------------------------------------------------------------------------*/
 /*  Scattering : rotate to meridian plane then scatter
/*-------------------------------------------------------------------------*
                                updateU(U, phi, theta, U2);  /* update photon trajectory */
                                costheta=cos(theta);
                                rotSphi(S, phi, S2);

                                S[0]= s11[ithedeg]*S2[0]+s12[ithedeg]*S2[1];
                                S[1]= s12[ithedeg]*S2[0]+s11[ithedeg]*S2[1];
                                S[2]= s33[ithedeg]*S2[2]+s43[ithedeg]*S2[3];
                                S[3]= -s43[ithedeg]*S2[2]+s33[ithedeg]*S2[3];
```

```
                                temp=(sqrt(1-costheta*costheta)*sqrt(1-U2[2]*U2[2]));
                                if ( temp==0){
                                        cosi=0;}
                                else{
                                if ((phi>pi) & (phi<2*pi))
                                                cosi=(U2[2]*costheta-U[2])/temp;
                                        else
                                                cosi=-(U2[2]*costheta-U[2])/temp;
                                if (cosi<-1.0) cosi=-1.0;
                                if (cosi>1.0) cosi=1.0;
                                }
                                sini = sqrt(1-cosi*cosi);
                                cos22=2*cosi*cosi-1;
                                sin22=2*sini*cosi;

                                S2[0]=S[0];
                                S2[1]=(S[1]*cos22-S[2]*sin22);
                                S2[2]=(S[1]*sin22+S[2]*cos22);
                                S2[3]=S[3];


                                S[1]= S2[1]/S2[0];
                                S[2]= S2[2]/S2[0];
                                S[3]= S2[3]/S2[0];
                                S[0]= 1.0;

                                for (i=0; i<3; i++) U[i] = U2[i]; /* update U */

/*ROULETTE*/
                                rnd=0; while(rnd==0) rnd=RandomNum;

                                if (W<THRESHOLD){
                                        if (rnd<=CHANCE)
                                                W/=CHANCE;
                                        else photon_status=DEAD;
                                }
        } /* end of single photon launching */

}/* slab size*/

printf("R= %5.5f\t %5.5f\t %5.5f\t %5.5f\n ",IR_1/(Nphotons-totcnt),QR_1/(Nphotons-
                totcnt),UR_1/(Nphotons-totcnt),VR_1/(Nphotons-totcnt));
printf("T= %5.5f\t %5.5f\t %5.5f\t %5.5f\n ",IT/(Nphotons-totcnt),QT/(Nphotons-
                totcnt),UT/(Nphotons-totcnt),VT/(Nphotons-totcnt));
```

```
IT=0;
QT=0;
UT=0;
VT=0;

IR_1=0;
QR_1=0;
UR_1=0;
VR_1=0;
/* H*/
if (jjj==1) {
        target = fopen("outHI.dat","w");

        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", IR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", IR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);

        target = fopen("outHQ.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", QR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", QR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outHU.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", UR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", UR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);

        target = fopen("outHV.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", VR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", VR[i][j]);
                fprintf(target,"\n");
```

```
                }
        fclose(target);

        for (iy=0; iy<NN; iy++)
                for (ix=0; ix<NN; ix++) {
                        IR[iy][ix] = 0.0;
                        QR[iy][ix] = 0.0;
                        UR[iy][ix] = 0.0;
                        VR[iy][ix] = 0.0;
                }
        }
/*V*/

        if (jjj==2) {
/* save data to file */
        target = fopen("outVI.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", IR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", IR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outVQ.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", QR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", QR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outVU.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", UR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", UR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outVV.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", VR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", VR[i][j]);
```

```
                fprintf(target,"\n");
                }
        fclose(target);
        for (iy=0; iy<NN; iy++)
                for (ix=0; ix<NN; ix++) {
                        IR[iy][ix] = 0.0;
                        QR[iy][ix] = 0.0;
                        UR[iy][ix] = 0.0;
                        VR[iy][ix] = 0.0;
                        }
        }
/*P*/
        if (jjj==3) {
/* save data to file */
        target = fopen("outPI.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", IR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", IR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outPQ.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", QR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", QR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outPU.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", UR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", UR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outPV.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", VR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", VR[i][j]);
                fprintf(target,"\n");
```

```
                }
        fclose(target);
        for (iy=0; iy<NN; iy++)
                for (ix=0; ix<NN; ix++) {
                        IR[iy][ix] = 0.0;
                        QR[iy][ix] = 0.0;
                        UR[iy][ix] = 0.0;
                        VR[iy][ix] = 0.0;
                }
        }
/* Q*/
        if (jjj==4) {
/* save data to file */
        target = fopen("outRI.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", IR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", IR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outRQ.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", QR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", QR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outRU.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", UR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", UR[i][j]);
                fprintf(target,"\n");
                }
        fclose(target);
        target = fopen("outRV.dat","w");
        for (i=0; i<NN; i++) {
                fprintf(target,"%5.5f", VR[i][0]);
                for (j=1; j<NN; j++)
                        fprintf(target,"\t%5.5f", VR[i][j]);
                fprintf(target,"\n");
                }
```

```
            fclose(target);
            }
}/* end of 4 photon launchings */
            finish_time= clock();
            printf("Elapsed Time = %10.2f seconds\n", (double)(finish_time-
                        start_time)/CLOCKS_PER_SEC);
            fflush(NULL);
return 0;
} /* main routine*/

/*************************************************************/
/* SUBROUTINES */
/*****************************************************************************
***
 *          RandomGen
 *     A random number generator that generates uniformly
 *     distributed random numbers between 0 and 1 inclusive.
 *     The algorithm is based on:
 *     W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P.
 *     Flannery, "Numerical Recipes in C," Cambridge University
 *     Press, 2nd edition, (1992).
 *     and
 *     D.E. Knuth, "Seminumerical Algorithms," 2nd edition, vol. 2
 *     of "The Art of Computer Programming", Addison-Wesley, (1981).
 *
 *     When Type is 0, sets Seed as the seed. Make sure 0<Seed<32000.
 *     When Type is 1, returns a random number.
 *     When Type is 2, gets the status of the generator.
 *     When Type is 3, restores the status of the generator.
 *
 *     The status of the generator is represented by Status[0..56].
 *
 *     Make sure you initialize the seed before you get random
 *     numbers.
 ****/
#define MBIG 1000000000
#define MSEED 161803398
#define MZ 0
#define FAC 1.0E-9

double
RandomGen(char Type, long Seed, long *Status){
  static long i1, i2, ma[56];   /* ma[0] is not used. */
  long        mj, mk;
```

```
short       i, ii;

if (Type == 0) {              /* set seed. */
  mj = MSEED - (Seed < 0 ? -Seed : Seed);
  mj %= MBIG;
  ma[55] = mj;
  mk = 1;
  for (i = 1; i <= 54; i++) {
    ii = (21 * i) % 55;
    ma[ii] = mk;
    mk = mj - mk;
    if (mk < MZ)
      mk += MBIG;
    mj = ma[ii];
  }
  for (ii = 1; ii <= 4; ii++)
    for (i = 1; i <= 55; i++) {
      ma[i] -= ma[1 + (i + 30) % 55];
      if (ma[i] < MZ)
        ma[i] += MBIG;
    }
  i1 = 0;
  i2 = 31;
} else if (Type == 1) {       /* get a number. */
  if (++i1 == 56)
    i1 = 1;
  if (++i2 == 56)
    i2 = 1;
  mj = ma[i1] - ma[i2];
  if (mj < MZ)
    mj += MBIG;
  ma[i1] = mj;
  return (mj * FAC);
} else if (Type == 2) {       /* get status. */
  for (i = 0; i < 55; i++)
    Status[i] = ma[i + 1];
  Status[55] = i1;
  Status[56] = i2;
} else if (Type == 3) {       /* restore status. */
  for (i = 0; i < 55; i++)
    ma[i + 1] = Status[i];
  i1 = Status[55];
  i2 = Status[56];
} else
```

```
   puts("Wrong parameter to RandomGen().");
  return (0);
}
#undef MBIG
#undef MSEED
#undef MZ
#undef FAC

/**********************************************************************
 *          rotSphi(S,phi,S)
 *                  Rotate S by phi [radians] and return as S
 *     multiply S for the rotational matrix of Chandrasekar or Boheren and Hoffman
 *                  Uses invtan()
 ****/
void        rotSphi(double* S, double phi, double* S2) {
            double     cos2phi, sin2phi;

            cos2phi = cos(2*phi);
            sin2phi = sin(2*phi);
            S2[0] = S[0];
            S2[1] = S[1]*cos2phi+S[2]*sin2phi;
            S2[2] = -S[1]*sin2phi+S[2]*cos2phi;
            S2[3] = S[3];
}
/**********************************************************************
*           updateU(U,U2)
****/
void        updateU(double* U, double phi, double theta, double* U2) {
            double     ux, uy, uz, uxx, uyy, uzz, temp, sintheta, costheta, sinphi, cosphi;
            double     pi = 3.14159265358979;
            ux = U[0];
            uy = U[1];
            uz = U[2];

            costheta = cos(theta);
            sintheta = sqrt(1.0 - costheta*costheta); /* sqrt() is faster than sin(). */
            cosphi   = cos(phi);
            if (phi < pi)
                        sinphi = sqrt(1.0 - cosphi*cosphi);     /* sqrt() is faster than sin(). */
            else
                        sinphi = -sqrt(1.0 - cosphi*cosphi);

 /* New directional cosines. */
 if (1 - fabs(uz) <= 1.0E-12) {      /* close to perpendicular. */
```

```
            uxx = sintheta * cosphi;
            uyy = sintheta * sinphi;
                        uzz = costheta * SIGN(uz);   /* is faster than division. */
            }
 else {
            temp = sqrt(1.0 - uz * uz);
            uxx = sintheta * (ux * uz * cosphi - uy * sinphi) / temp + ux * costheta;
            uyy = sintheta * (uy * uz * cosphi + ux * sinphi) / temp + uy * costheta;
            uzz = -sintheta * cosphi * temp + uz * costheta;
  }
 /* Update directional cosines */
 U2[0] = uxx;
 U2[1] = uyy;
 U2[2] = uzz;
}
```

B2. Euler Monte Carlo

```
/*****************************************************************
*
```

Copyright Jessica C. Ramella-Roman, Steve L. Jacques and Scott A. Prahl 2003 Stok1.c- Euler rotations: Main program for Monte Carlo simulation of photon travelling into scattering media keeping track of its status of polarization. Slab geometry.
by Jessica C. Ramella-Roman.
This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

```
****/

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "array.h"
#include "complex.h"
#include "mie.h"
#include "nrutil.h"
```

```
#include <time.h>

#define ALIVE       1
#define DEAD        0
#define    NN          100
#define THRESHOLD   0.01              /* used in roulette */
#define CHANCE      0.1               /* used in roulette */

#define RandomNum (double) RandomGen(1, 0, NULL)
#define SIGN(x) ((x)>=0 ? 1:-1)
#define InitRandomGen (double) RandomGen(0, 1, NULL)

/* Declare Subroutines */
double     RandomGen(char Type, long Seed, long *Status);
void       multS(double* S, double theta, double* S2);
void       rotSphi(double* S, double phi, double* S2);
void    rotateYYZZ(double* YY,double* ZZ, double phi, double theta, double* YY2,
double* ZZ2);

/********************************** MAIN
int main(void) {
           double pi = 3.1415926535897932384;

           /* Mie theory stuff */

           double radius,lambda, A;
           long nangles,i, j;
           struct complex m;
           struct complex*s1=NULL;
           struct complex*s2=NULL;
           double *mu=NULL;
           double x,qext,qsca,qback,g, rho, vol;
           double nre_p, nim_p, nre_med, nim_med;
           FILE *target;
           double     IR_1, QR_1, UR_1, VR_1;
           double     IT_1, QT_1, UT_1, VT_1;
           int maxcnt;

           /* E field*/
           double     phi, theta, I,I0;
           long       ithedeg;

           /* Propagation parameters */
           double     y,z,dy, dx, hw;/* photon position.*/
```

```
        double     s;          /* step sizes. s = -log(RND)/mus [cm] */
        long       i_photon;   /* current photon */
        short   photon_status;  /*  = ALIVE=1 or DEAD=0 */
        double     mua;        /* absorption coefficient [cm^-1] */
        double     mus;        /* scattering coefficient [cm^-1] */
        double  albedo;     /* albedo of tissue */
        int        cnt;
        double  rnd ;       /* assigned random value 0-1 */
        int     ix,iy, MM;
        double     absorb,W;

/**** allocate matrices and arrays *******/

        double     **IR, **QR, **UR, **VR;
        double     *U, *U2;
        double     *S;         /* */
        double     *S0;        /* */
        double     *S2;        /* */
        double     *s11=NULL;
        double     *s12=NULL;
        double     *s33=NULL;
        double     *s43=NULL;
        double slabsize;
        int        jjj;
        int                    Nphotons,totcnt;
        double     *XX, *XX2,*YY, *YY2,*ZZ, *ZZ2;
        double     *IQUV;              /* [I, Q, U, V] Stokes Vector */
        double      start_time, finish_time;

        MM = NN - 1;
        U      = new_darray(3);
        U2     = new_darray(3);
        S      = new_darray(4);
        S0     = new_darray(4);
        S2     = new_darray(4);/* dummy S*/

        XX     = new_darray(3);
        YY     = new_darray(3);
        ZZ     = new_darray(3);

        XX2     = new_darray(6);
        YY2     = new_darray(6);
        ZZ2     = new_darray(6);
        IQUV    = new_darray(4);
```

```
        start_time = clock();

/* Choose MIE SCATTERING parameters */
        radius          = 2.0/2; /* microns */
        lambda          = 0.6328; /* microns */
        rho             = 1.152e-4;/*Dilution 1*/
        Nphotons        = 1e6;
        mua             = 0.0; /*μa  */
        nre_p           = 1.59;
        nim_p           = 0;
        nre_med         = 1.333;
        nim_med         = 0.0;
        nangles         = 10000;

        IR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
        QR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
        UR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
        VR    = dmatrix(0, MM, 0, MM); /* [0:MM] */

/* Setup MIE SCATTERING parameters */
        mu  = new_darray(nangles);
        s1  = new_carray(nangles);
        s2  = new_carray(nangles);

        m.re = nre_p/nre_med;
        m.im = 0.0;
        x    = 2*pi*radius/(lambda/nre_med);
        vol  = 4.0/3*pi*radius*radius*radius;
        A    = pi*radius*radius;
        for(i=0;i<nangles;i++)
        mu[i] = cos(pi*i/nangles);
        s11=new_darray(nangles);
        s12=new_darray(nangles);
        s33=new_darray(nangles);
        s43=new_darray(nangles);
        s1=new_carray(nangles);
        s2=new_carray(nangles);

        Mie(x,m,mu,nangles,s1,s2,&qext,&qsca,&qback,&g); /* Mie program ----- */
        mus  = qsca*A*rho*1e4;
        albedo = mus/(mus + mua);
        free_darray(mu);
```

```
        for(i=0;i<nangles;++i){
                s11[i] = 0.5*cabbs(s2[i])*cabbs(s2[i]) +
                        0.5*cabbs(s1[i])*cabbs(s1[i]);
                s12[i] = 0.5*cabbs(s2[i])*cabbs(s2[i]) –
                        0.5*cabbs(s1[i])*cabbs(s1[i]);
                s33[i] = (cmul(conj(s1[i]),s2[i])).re;
                s43[i] = (cmul(conj(s1[i]),s2[i])).im;
        }
        IR_1=0;/*W*/
        QR_1=0;
        UR_1=0;
        VR_1=0;
        IT_1=0;/*W*/
        QT_1=0;
        UT_1=0;
        VT_1=0;

        for (iy=0; iy<NN; iy++)
                for (ix=0; ix<NN; ix++) {
                        IR[iy][ix] = 0.0;
                        QR[iy][ix] = 0.0;
                        UR[iy][ix] = 0.0;
                        VR[iy][ix] = 0.0;
                        }

/* SET UP Monte Carlo */
        hw                      = 7/mus; /*
        dx                      = 2.0*hw/NN;
        dy                      = 2.0*hw/NN;
        maxcnt                  = 50;
        slabsize                = 4/mus;
        InitRandomGen;

/* LAUNCHNOW*/
        for (jjj = 1; jjj <= 4; jjj++) {
                if (jjj == 1){
                        S0[0] = 1;
                        S0[1] = 1;
                        S0[2] = 0;
                        S0[3] = 0;
                        printf("launch H\n");
                        }

                if (jjj == 2){
```

```
                    S0[0] = 1;
                    S0[1] =    -1;
                    S0[2] = 0;
                    S0[3] = 0;
                    printf("launch V\n");}

                if (jjj == 3){
                    S0[0] = 1;
                    S0[1] =    0;
                    S0[2] = 1;
                    S0[3] = 0;
                    printf("launch P\n");}

                if (jjj == 4){
                    S0[0] = 1;
                    S0[1] =    0;
                    S0[2] = 0;
                    S0[3] = 1;
                    printf("launch R\n");}
                    totcnt = 0;

/* LAUNCH photon */
                for (i_photon = 1; i_photon <= Nphotons; i_photon++) {

/* photon position */
                    x = 0.0;
                    y = 0.0;
                    z = 0.0;
                    XX[0] = 1.0; /* original X-axis coord */
                    XX[1] = 0.0;
                    XX[2] = 0.0;
                    YY[0] = 0.0; /* original Y-axis coord */
                    YY[1] = 1.0;
                    YY[2] = 0.0;
                    ZZ[0] = 0.0; /* original Y-axis coord */
                    ZZ[1] = 0.0;
                    ZZ[2] = 1.0;
                    rotateYYZZ(YY,XX,phi, theta, YY2,XX2
                    rotateYYZZ(ZZ,XX,phi, theta, ZZ2,XX2);
                    for (i=0; i<3; i++) {
                        YY[i] = YY2[i];
                        ZZ[i] = ZZ2[i];
                    }
                    for (i=0; i<4; i++) S[i] = S0[i];
```

```
                        for (i=0; i<4; i++) S2[i] = 0.0;
                        cnt = 0;
                        photon_status = ALIVE;
                        W           = 1; /* photon weight */
                        while (photon_status == ALIVE) {
                        cnt+=1;
                        if (cnt>maxcnt){
                                photon_status = DEAD;
                        totcnt+=1;}
/**** HOP */
                rnd = 0; while (rnd == 0) rnd = RandomNum;  /* choose a step size
*/

                s = -log(RandomNum)/(mus+mua);

                x += ZZ[0]*s;
                y += ZZ[1]*s;
                z += ZZ[2]*s;

/**** ABSORB */

                absorb = W*(1-albedo);
                W-= absorb;

                if (z <= 0) {

                        XX[1]=-(ZZ[0]*YY[2]-ZZ[2]*YY[0]);
                        XX[2]=(ZZ[0]*YY[1]-ZZ[1]*YY[0]);
                        if (fabs(YY[2])<1e-8 && fabs(XX[2])<1e-8){
                }
                else {
                        phi = atan2(YY[2],-XX[2]);
                        rotSphi(S, phi, S2);
                        phi= atan2(ZZ[1],ZZ[0]);
                        rotSphi(S2, phi, S);
                        }
                IR_1+=S[0];
                QR_1+=S[1];
                UR_1+=S[2];
                VR_1+=S[3];
                if (x >= -hw)
                        ix = (int)(fabs(x + hw)/dx);
                if (y >= -hw)
```

```
                iy = (int)(fabs(y + hw)/dy);
        if (ix > MM) ix = MM; /* last bin [MM] is overflow */
        if (iy > MM) iy = MM; /* last bin [MM] is overflow */
        IR[iy][ix] += S[0];
        QR[iy][ix] += S[1];
        UR[iy][ix] += S[2];
        VR[iy][ix] += S[3];
        photon_status = DEAD;
        }
        else if ( z>=slabsize) {
                XX[1]=-(ZZ[0]*YY[2]-ZZ[2]*YY[0]);
                XX[2]=(ZZ[0]*YY[1]-ZZ[1]*YY[0]);
        if (fabs(YY[2])<1e-8 && fabs(XX[2])<1e-8){
                }
        else {
                phi = atan2(YY[2],-XX[2]);
                rotSphi(S, phi, S2);
                phi= atan2(ZZ[1],-ZZ[0]);
                rotSphi(S2, phi, S);
        }
        IT_1+=S[0];
        QT_1+=S[1];
        UT_1+=S[2];
        VT_1+=S[3];
        photon_status = DEAD;

        }/*z>slab size*/

/* REJECTION METHOD to choose azimuthal angle phi and deflection angle theta */

        do{ theta = acos(2*RandomNum-1);
                phi = RandomNum*2.0*pi;
        I0=s11[0]*S[0]+s12[0]*(S[1]*cos(2*phi)+S[2]*sin(2*phi));
                ithedeg = floor(theta*180/pi*nangles/180);
                I=s11[ithedeg
                ]*S[0]+s12[ithedeg]*(S[1]*cos(2*phi)+S[2]*sin(2*phi));

        }while(RandomNum*I0>=I);
                rotSphi(S, phi, S2);
                for (i=0; i<4; i++) S[i] = S2[i];
                S2[0]= s11[ithedeg]*S[0]+s12[ithedeg]*S[1];
                S2[1]= s12[ithedeg]*S[0]+s11[ithedeg]*S[1];
                S2[2]= s33[ithedeg]*S[2]+s43[ithedeg]*S[3];
```

```
                                S2[3]= -s43[ithedeg]*S[2]+s33[ithedeg]*S[3];
                                S[1]= S2[1]/S2[0];
                                S[2]= S2[2]/S2[0];
                                S[3]= S2[3]/S2[0];
                                S[0]= 1.0;
/* rotate coord of original x and y axis in local photon frame */
                                rotateYYZZ(YY,ZZ,phi, theta, YY2,ZZ2);
                                for (i=0; i<3; i++) {
                                YY[i] = YY2[i];
                                ZZ[i] = ZZ2[i];
                                }
/*ROULETTE*/
                                rnd=0; while(rnd==0) rnd=RandomNum;
                                if (W<THRESHOLD){
                                if (rnd<=CHANCE)
                                W/=CHANCE;
                                else photon_status=DEAD;
                                }
                        } /* while (photon_status == ALIVE) */

} /* end of single photon launching */

IR_1=0;
QR_1=0;
UR_1=0;
VR_1=0;

IT_1=0;
QT_1=0;
UT_1=0;
VT_1=0;

/*Place OUT ROUTINE #1 here*/

}/* end of 4 photon launchings */
            finish_time= clock();
            printf("Elapsed Time = %10.2f seconds\n", (double)(finish_time-
            start_time)/CLOCKS_PER_SEC);
}/*main*/

/*************************************************************/
/* SUBROUTINES */
```

```
/*****************
/*          Place RandomGen routine here/*
/************************************************/
*           rotSphi(S,phi,S)
 *          Rotate S by phi [radians] and return as S multiply S for the rotational matrix
of *        Chandrasekar or Boheren and Hoffman
 ****/
void        rotSphi(double* S, double phi, double* S2) {
            double     cos2phi, sin2phi;

            cos2phi = cos(2*phi);
            sin2phi = sin(2*phi);

            S2[0] = S[0];
            S2[1] = S[1]*cos2phi+S[2]*sin2phi;
            S2[2] = -S[1]*sin2phi+S[2]*cos2phi;
            S2[3] = S[3];
}

/*************************************************************************
*           rotateYYZZ(YY,ZZ,phi, theta, YY,ZZ)
*           Rotates coordinates of original x and y coord as seen by local photon frame.
****/
void        rotateYYZZ(double* YY,double* ZZ, double phi, double theta, double* YY2,
             double* ZZ2) {
            double ct,st,vt,temp;
            int i;
            double     pi = 3.14159265358979;

            /* phi rotation around local z axis */
            ct = cos(phi);
            st = sin(phi);
            vt = 1-ct;

            YY2[0]  = (ZZ[0]*ZZ[0]*vt+ct)*YY[0]+(ZZ[0]*ZZ[1]*vt-
            ZZ[2]*st)*YY[1]+(ZZ[0]*ZZ[2]*vt+ZZ[1]*st)*YY[2];
            YY2[1]  =(ZZ[0]*ZZ[1]*vt+ZZ[2]*st)*YY[0]+
            (ZZ[1]*ZZ[1]*vt+ct)*YY[1]+(ZZ[1]*ZZ[2]*vt-ZZ[0]*st)*YY[2];
            YY2[2]  = (ZZ[0]*ZZ[2]*vt-
            ZZ[1]*st)*YY[0]+(ZZ[1]*ZZ[2]*vt+
            ZZ[0]*st)*YY[1]+(ZZ[2]*ZZ[2]*vt+ct)*YY[2];

            temp=1/sqrt(YY2[0]*YY2[0]+YY2[1]*YY2[1]+YY2[2]*YY2[2]);
```

```
        YY2[0]=YY2[0]*temp;
        YY2[1]=YY2[1]*temp;
        YY2[2]=YY2[2]*temp;

        ct = cos(theta);
        st = sin(theta);
        vt = 1-ct;

        ZZ2[0]  = (YY2[0]*YY2[0]*vt+ct)*ZZ[0]+(YY2[0]*YY2[1]*vt-
        YY2[2]*st)*ZZ[1]+(YY2[0]*YY2[2]*vt+YY2[1]*st)*ZZ[2];
        ZZ2[1]  =(YY2[0]*YY2[1]*vt+YY2[2]*st)*ZZ[0]+(YY2[1]*YY2[1]*vt
        +ct)*ZZ[1]+(YY2[1]*YY2[2]*vt-YY2[0]*st)*ZZ[2];
        ZZ2[2]  = (YY2[0]*YY2[2]*vt-YY2[1]*st)*ZZ[0]+(YY2[1]*YY2[2]*vt+
        YY2[0]*st)*ZZ[1]+(YY2[2]*YY2[2]*vt+ct)*ZZ[2];

        temp=1/fabs(sqrt(ZZ2[0]*ZZ2[0]+ZZ2[1]*ZZ2[1]+ZZ2[2]*ZZ2[2]));
        ZZ2[0]=ZZ2[0]*temp;
        ZZ2[1]=ZZ2[1]*temp;
        ZZ2[2]=ZZ2[2]*temp;
}
```

B3. Quaternions Monte Carlo

```
/*****************************************************************
```

Copyright Jessica C. Ramella-Roman, Steve L. Jacques and Scott A. Prahl 2003 Stok1.c- Quaternions Monte Carlo: Main program for Monte Carlo simulation of photon travelling into scattering media keeping track of its status of polarization. Slab geometry.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

```
****/
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "array.h"
```

```
#include "complex.h"
#include "mie.h"
#include "nrutil.h"
#include <time.h>

#define ALIVE       1
#define DEAD        0
#define    NN          100
#define THRESHOLD   0.01              /* used in roulette */
#define CHANCE      0.1               /* used in rou lette */

#define RandomNum (double) RandomGen(1, 0, NULL)
#define SIGN(x) ((x)>=0 ? 1:-1)
#define InitRandomGen (double) RandomGen(0, 1, NULL)

/* Declare Subroutines */
void        rotSphi(double* S, double phi, double* S2);
double      RandomGen(char Type, long Seed, long *Status);
void        multS(double* S, double theta, double* S2);
void        updateU(double* U, double phi, double theta, double* U2);
void    multq(double* Q, double* dd, double* cc);
void        makeq(double* v, double phi, double* Q);

/* MAIN */
int main() {
            double pi = 3.1415926535897932384;
            /* Mie theory stuff */
            double radius,lambda, A;
            long nangles,i;
            struct complex m;
            struct complex*s1=NULL;
            struct complex*s2=NULL;
            double *mu=NULL;
            double x,qext,qsca,qback,g, rho, vol,dy, dx, hw;
            double nre_p, nim_p, nre_med, nim_med;
            FILE *target;
            double jjj;
            /* E field stuff */
            double      phi, theta,I,I0;
            int         ithedeg;
            double      IT_1, QT_1, UT_1, VT_1;
            double      IR_1, QR_1, UR_1, VR_1;
            double       start_time, finish_time;
            double      *XX,*YY,*ZZ,*XX2,*YY2,*ZZ2,*Q;
```

```
            double    **IR, **QR, **UR, **VR;
            double    y,z;
            double    s;
            long      i_photon;   /* current photon */
            int       Nphotons;   /* number of photons in simulation */
            short   photon_status;  /*  = ALIVE=1 or DEAD=0 */

            /* other variables */
            double  absorb;
            double    mua;      /* absorption coefficient [cm^-1] */
            double    mus;      /* scattering coefficient [cm^-1] */
            double    musp;     /* reduced scattering coefficient [cm^-1] */
            double  albedo;    /* albedo of tissue */
            /* dummy variables */
            int       maxcnt,cnt,totcnt;
            double  rnd;      /* assigned random value 0-1 */
            int    MM;
            double  W;        /* photon weight */
            double  slabsize;
            int       j,ix,iy;

/**** allocate matrices and arrays *******/
            double    *U, *U2;
            double    *S;       /* */
            double    *S0;      /* */
            double    *S2;      /* */
            double    *s11=NULL;
            double    *s12=NULL;
            double    *s33=NULL;
            double    *s43=NULL;
            double    *IQUV;            /* [I, Q, U, V] Stokes Vector */
            double    temp;
start_time = clock();
MM = NN - 1;
U     = new_darray(3);
U2    = new_darray(3);
S     = new_darray(4);
S0    = new_darray(4);
S2    = new_darray(4);/* dummy S*/
IQUV  = new_darray(4);

IR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
QR    = dmatrix(0, MM, 0, MM); /* [0:MM] */
```

```
UR     = dmatrix(0, MM, 0, MM); /* [0:MM] */
VR     = dmatrix(0, MM, 0, MM); /* [0:MM] */

XX     = new_darray(3);
ZZ     = new_darray(3);
YY     = new_darray(3);
YY2    = new_darray(3);
XX2    = new_darray(3);
ZZ2    = new_darray(3);
Q      = new_darray(4);


/* Choose MIE SCATTERING parameters */
radius          = 2.0/2; /* microns */
lambda          = 0.6328; /* microns */
rho             = 1.152e-4;/*Dilution 1*/
Nphotons        = 1e5;
mua             = 0.0; /*μa  */
nre_p           = 1.59;
nim_p           = 0;
nre_med         = 1.333;
nim_med         = 0.0;
nangles         = 10000;

/* Setup MIE SCATTERING parameters */

mu  = new_darray(nangles);
s1  = new_carray(nangles);
s2  = new_carray(nangles);
s11 = new_darray(nangles);
s12 = new_darray(nangles);
s33 = new_darray(nangles);
s43 = new_darray(nangles);

m.re = nre_p/nre_med;
m.im = 0.0;
x    = 2*pi*radius/(lambda/nre_med);
vol  = 4.0/3*pi*radius*radius*radius;
A    = pi*radius*radius;

for(i=0;i<nangles;i++)
        mu[i] = cos(pi*i/nangles);
        Mie(x,m,mu,nangles,s1,s2,&qext,&qsca,&qback,&g); /* Mie program */
```

```
        mus     = qsca*A*rho*1e4; /* [cm^-1] */
        albedo  = mus/(mus + mua);
        free_darray(mu);


for(i=0;i<nangles;++i){
        s11[i] = 0.5*cabbs(s2[i])*cabbs(s2[i]) + 0.5*cabbs(s1[i])*cabbs(s1[i]);
        s12[i] = 0.5*cabbs(s2[i])*cabbs(s2[i]) - 0.5*cabbs(s1[i])*cabbs(s1[i]);
        s33[i] = (cmul(conj(s1[i]),s2[i])).re;
        s43[i] = (cmul(conj(s1[i]),s2[i])).im;
        */}
        hw      = 0.8; /* [cm] , maximum range in x and y for output. */
        dx      = 2.0*hw/NN;
        dy      = 2.0*hw/NN;
        maxcnt  = 50; /*kill photon after maxcnt scattering events*/

/******** MONTE CARLO *******/

        InitRandomGen;

/* LAUNCHNOW*/
        IT_1=0;
        QT_1=0;
        UT_1=0;
        VT_1=0;

        IR_1=0;
        QR_1=0;
        UR_1=0;
        VR_1=0;

        temp=0;

        for (iy=0; iy<NN; iy++){
                for (ix=0; ix<NN; ix++) {
                        IR[iy][ix] = 0.0;
                        QR[iy][ix] = 0.0;
                        UR[iy][ix] = 0.0;
                        VR[iy][ix] = 0.0;
                        }
        }

        slabsize = 4.0/mus;
```

```
    for (jjj = 1; jjj <= 4; jjj++) {
                        if (jjj == 1){
                                S0[0] = 1.0;
                                S0[1] = 1.0 ;
                                S0[2] = 0.0;
                                S0[3] = 0.0;
                                printf("launch H\n");}

                        if (jjj == 2){
                                S0[0] = 1.0;
                                S0[1] =-1.0;
                                S0[2] = 0.0;
                                S0[3] = 0.0;
                                printf("launch V\n");}

                        if (jjj == 3){

                                S0[0] = 1.0;
                                S0[1] =    0.0;
                                S0[2] = 1.0;
                                S0[3] = 0.0;
                                printf("launch P\n");}

                        if (jjj == 4){
                                S0[0] = 1.0;
                                S0[1] =    0.0;
                                S0[2] = 0.0;
                                S0[3] = 1.0;
                                printf("launch R\n");}
                        totcnt=0;

/* LAUNCH photon */
                        for (i_photon = 1; i_photon <= Nphotons; i_photon++) {

                        x = 0.0;
                        y = 0.0;
                        z = 0.0;

                        U[0]=0.0;
                        U[1]=0.0;
                        U[2]=1.0;

                        XX[0] = 1.0; /* original X-axis coord */
```

```
                XX[1] = 0.0;
                XX[2] = 0.0;

                YY[0] = 0.0; /* original Y-axis coord */
                YY[1] = 1.0;
                YY[2] = 0.0;

                ZZ[0] = 0.0; /* original Z-axis coord */
                ZZ[1] = 0.0;
                ZZ[2] = 1.0;

                theta=-56*pi/180;
/*Quaternion creation and multiplication*/
                makeq(XX, theta, Q);
                multq(Q,YY,YY2);
                multq(Q,ZZ,ZZ2);

                for (i=0; i<3; i++){
                        YY[i] = YY2[i];
                        ZZ[i] = ZZ2[i];
                }

                for (i=0; i<4; i++) S[i] = S0[i]; /* set incident Stokes vector to S0 */
                cnt = 0;
                photon_status = ALIVE;
                W =1;

/********* ALIVE cycle *****************/
                while (photon_status == ALIVE) {
                        cnt+=1;
                        if (cnt>maxcnt){
                                photon_status = DEAD;
                                totcnt+=1;
                        }

/**** HOP */
        rnd = 0; while (rnd == 0) rnd = RandomNum;  /* choose a step size */
                        s = -log(rnd)/(mus+mua);
                        x += ZZ[0]*s;
                        y += ZZ[1]*s;
                        z += ZZ[2]*s;

/**** ABSORB */
                        absorb = W*(1-albedo);
```

```
                                W-= absorb;
if ( z<=0) {
        XX[1]=-(ZZ[0]*YY[2]-ZZ[2]*YY[0]);
        XX[2]=(ZZ[0]*YY[1]-ZZ[1]*YY[0]);
        if (fabs(YY[2])<1e-8 && fabs(XX[2])<1e-8){
        }
        else{
                phi = atan2(YY[2],-XX[2]);
                rotSphi(S, phi, S2);
                phi= atan2(ZZ[1],ZZ[0]);
                rotSphi(S2, phi, S);
        }

        IR_1+=S[0];
        QR_1+=S[1];
        UR_1+=S[2];
        VR_1+=S[3];

        if (x >= -hw)
                ix = (int)(fabs(x + hw)/dx);
        if (y >= -hw)
                iy = (int)(fabs(y + hw)/dy);
        if (ix > MM) ix = MM; /* last bin [MM] is overflow */
        if (iy > MM) iy = MM; /* last bin [MM] is overflow */

        IR[iy][ix] += S[0];
        QR[iy][ix] += S[1];
        UR[iy][ix] += S[2];
        VR[iy][ix] += S[3];
        photon_status = DEAD;
        }

else if ( z>=slabsize) {
        XX[1]=-(ZZ[0]*YY[2]-ZZ[2]*YY[0]);
        XX[2]=(ZZ[0]*YY[1]-ZZ[1]*YY[0]);
        if (fabs(YY[2])<1e-8 && fabs(XX[2])<1e-8){
        }
        else {
                phi = atan2(YY[2],-XX[2]);
                rotSphi(S, phi, S2);
                phi= atan2(ZZ[1],-ZZ[0]);
                rotSphi(S2, phi, S);
        }
```

```
            IT_1+=S[0];
            QT_1+=S[1];
            UT_1+=S[2];
            VT_1+=S[3];
            photon_status = DEAD;
}/*z>slab size*/

/* SPIN */
/* REJECTION METHOD to choose azimuthal angle phi and deflection angle theta */

            do{ theta = acos(2*RandomNum-1);
                        phi = RandomNum*2.0*pi;
                        I0=s11[0]*S[0]+s12[0]*(S[1]*cos(2*phi)+S[2]*sin(2*phi));
                        ithedeg = floor(theta*180/pi*nangles/180);
                        I=s11[ithedeg ]*S[0]+s12[ithedeg]*(S[1]*cos(2*phi)+
                        S[2]*sin(2*phi));
            }while(RandomNum*I0>=I);

            rotSphi(S, phi, S2);

            S[0]= s11[ithedeg]*S2[0]+s12[ithedeg]*S2[1];
            S[1]= s12[ithedeg]*S2[0]+s11[ithedeg]*S2[1];
            S[2]= s33[ithedeg]*S2[2]+s43[ithedeg]*S2[3];
            S[3]= -s43[ithedeg]*S2[2]+s33[ithedeg]*S2[3];

            S[1]= S[1]/S[0];
            S[2]= S[2]/S[0];
            S[3]= S[3]/S[0];
            S[0]= 1.0;
            makeq(ZZ, phi, Q);
            multq(Q,YY,YY2);
            makeq(YY2, theta, Q);
            multq(Q,ZZ,ZZ2);
            for (i=0; i<3; i++){
            ZZ[i] = ZZ2[i];
            YY[i] = YY2[i];}

            /*ROULETTE*/
            rnd=0; while(rnd==0) rnd=RandomNum;

            if (W<THRESHOLD){
                        if (rnd<=CHANCE)
                                    W/=CHANCE;
```

```
                                else photon_status=DEAD;
                        }
                } /* end of single photon launching */

}/* slab size*/

IT_1=0;
QT_1=0;
UT_1=0;
VT_1=0;

IR_1=0;
QR_1=0;
UR_1=0;
VR_1=0;

                /*Place OUT ROUTINE #1 here*/

}/* end of 4 photon launchings */
                finish_time= clock();
                printf("Elapsed Time=%10.2f seconds\n", (double)(finish_time
                start_time)/CLOCKS_PER_SEC);
                fflush(NULL);
                W=1;
                return 0;
} /* main routine*/
/*************************************************************/
/* SUBROUTINES */
/*************************************************************

/* Place RandomGen here*/

/*************
 *              rotSphi(S,phi,S)
 •              Rotate S by phi [radians] and return as S multiply S for the
 •              rotational matrix of Chandrasekar or Boheren and Hoffman
****/
void            rotSphi(double* S, double phi, double* S2) {
                double     cos2phi, sin2phi;

                cos2phi = cos(2*phi);
                sin2phi = sin(2*phi);
                S2[0] = S[0];
```

```
        S2[1] = S[1]*cos2phi+S[2]*sin2phi;
        S2[2] = -S[1]*sin2phi+S[2]*cos2phi;
        S2[3] = S[3];
}

/*****************************************************************
 *      updateU(U,U2)
 ****/
void    updateU(double* U, double phi, double theta, double* U2) {
        double  ux, uy, uz, uxx, uyy, uzz, temp, sintheta, costheta, sinphi, cosphi;
        double  pi =  3.1415926535897932384;

        ux = U[0];
        uy = U[1];
        uz = U[2];

        costheta = cos(theta);
        sintheta = sin(theta);
        cosphi   = cos(phi);
        if (phi < pi)
                sinphi = sqrt(1.0 - cosphi*cosphi);     /* sqrt() is faster than sin(). */
        else
                sinphi = -sqrt(1.0 - cosphi*cosphi);

 /* New directional cosines. */
 if (1 - fabs(uz) <= 1.0E-12) {/* close to perpendicular. */
   uxx = sintheta * cosphi;
   uyy = sintheta * sinphi;
   uzz = costheta * SIGN(uz);   /* is faster than division. */
   }
 else {
   temp = sqrt(1.0 - uz * uz);
   uxx = sintheta * (ux * uz * cosphi - uy * sinphi) / temp + ux * costheta;
   uyy = sintheta * (uy * uz * cosphi + ux * sinphi) / temp + uy * costheta;
   uzz = -sintheta * cosphi * temp + uz * costheta;
   }
 /* Update directional cosines */
 U2[0] = uxx;
 U2[1] = uyy;
 U2[2] = uzz;
}

/*****************************************************************
*       makeq(XX,phi, Q)
```

```
*                       makes quaternion
****/
void        makeq(double* v, double phi1, double* Q) {
            double cosphi1, sinphi1,phi;

/* phi rotation around local z axis */
            cosphi1 = cos(phi1*0.5); /*phi1/2*/
            sinphi1 =  sin(phi1*0.5);

/*I included the conj ( - in the first three terms) */
            Q[0] = -v[0]*sinphi1;
            Q[1] = -v[1]*sinphi1;
            Q[2] = -v[2]*sinphi1;
            Q[3] = cosphi1;
}
/*****************************************************************
 *          multq(Q,XX, XX2)
 *          multiplies quaternion times the vector X,Y or Z
 ****/
void        multq(double* Q, double* dd, double* cc) {
double temp;

            cc[0] = -Q[0]*(-dd[0]*Q[0]-dd[1]*Q[1] -dd[2]*Q[2]) -Q[1]*(dd[0]*Q[1] –
            dd[1]*Q[0] + dd[2]*Q[3]) +Q[2]*(-dd[0]*Q[2] + dd[1]*Q[3] + dd[2]*Q[0]) +
            Q[3]*(dd[0]*Q[3] + dd[1]*Q[2] -dd[2]*Q[1]);

            cc[1] = +Q[0]*(dd[0]*Q[1] -dd[1]*Q[0] + dd[2]*Q[3]) -Q[1]*(-dd[0]*Q[0]-
            dd[1]*Q[1] -dd[2]*Q[2]) -Q[2]*(dd[0]*Q[3] + dd[1]*Q[2] -dd[2]*Q[1])+
            Q[3]*(-dd[0]*Q[2] + dd[1]*Q[3] + dd[2]*Q[0]);

            cc[2] = -Q[0]*(-dd[0]*Q[2] + dd[1]*Q[3] + dd[2]*Q[0])+Q[1]*(dd[0]*Q[3] +
            dd[1]*Q[2] -dd[2]*Q[1]) -Q[2]*(-dd[0]*Q[0]-dd[1]*Q[1] -dd[2]*Q[2]) +
            Q[3]*(dd[0]*Q[1] -dd[1]*Q[0] + dd[2]*Q[3]);

            temp  = 1/sqrt(cc[0]*cc[0]+cc[1]*cc[1]+cc[2]*cc[2]);
            cc[0] = cc[0]*temp;
            cc[1] = cc[1]*temp;
            cc[2] = cc[2]*temp;
}
```

Appendix C: Mueller Matrices of three tissue types

C.1 The Data Matrix recorded as transmission

The Data Matrix is listed in dimensionless units of transmission. The Data Matrix is called DM(ii, jj, i, j) where ii denotes the rows as the sources H, V, P and R, jj denotes the columns as the detectors H, V, P and R, i=1:5denotes the position on the sample and j=1:Ns denotes the sample thicknesses.Where a factor "1.0e-03 *" precedes the listed array of data, the data in that array should be scaled by that factor.

C.1.1 Skin (Ns = 3)

DM(:,:,1,1) =
0.00790.00180.00500.0046
0.00220.00840.00560.0049
0.00570.00540.01000.0054
0.00540.00540.00560.0072

DM(:,:,2,1) =
0.01010.00110.00600.0041
0.00120.01000.00580.0063
0.00620.00570.01100.0055
0.00690.00460.00620.0090

DM(:,:,3,1) =
0.01430.00400.00900.0132
0.00410.01350.00860.0035
0.00880.00850.01610.0081
0.00420.01390.00890.0122

DM(:,:,4,1) =
0.01790.00050.00920.0091
0.00060.01750.00980.0080
0.00930.00900.01810.0083
0.00860.00970.00930.0164

DM(:,:,5,1) =
0.02870.00160.02540.0215
0.00160.04780.02610.0263
0.02660.02390.04830.0276
0.02670.02250.02180.0435

DM(:,:,1,2) =
0.00230.00020.00140.0011
0.00020.00230.00120.0013
0.00160.00120.00200.0016
0.00140.00120.00090.0017

DM(:,:,2,2) =
0.00210.00060.00110.0015
0.00040.00190.00130.0003
0.00110.00130.00180.0015
0.00100.00160.00090.0012

DM(:,:,3,2) =
0.00290.00020.00210.0023
0.00020.00450.00280.0023
0.00260.00260.00330.0034
0.00260.00250.00160.0029

DM(:,:,4,2) =
0.00960.00050.00470.0047
0.00050.00900.00550.0047
0.00490.00530.00660.0060
0.00520.00500.00400.0058

DM(:,:,5,2) =
0.00460.00120.00150.0030
0.00120.00460.00420.0025
0.00160.00420.00250.0035
0.00230.00360.00230.0016

DM(:,:,1,3) =
1.0e-03 *
0.16940.07290.11950.1200
0.07600.17050.13030.1089

0.14390.11850.20450.1345
0.11620.15250.12760.1476

DM(:,:,2,3) =
1.0e-03 *
0.12540.09990.08290.1245
0.10540.12250.14510.0829
0.10570.12300.16310.1318
0.07080.16250.11120.0965

DM(:,:,3,3) =
1.0e-03 *
0.20680.06810.11590.1281
0.06930.19220.15130.1161
0.13100.13260.19120.1459
0.10690.14390.09760.1246

DM(:,:,4,3) =
1.0e-03 *
0.28550.10880.19100.1798
0.10990.27790.20100.1774
0.21780.18360.31030.1995
0.18330.21360.18620.2261

DM(:,:,5,3) =
1.0e-03 *
0.14970.03980.08900.0915
0.04100.15230.11330.0818
0.09150.10280.14860.0829
0.08010.10400.10060.0949

C.1.2 Muscle (Ns = 4)

DM(:,:,1,1) =
0.04370.01280.01030.0457
0.01250.06200.06100.0442
0.01080.06450.02760.0264
0.04190.03190.05700.0113

DM(:,:,2,1) =
0.03190.01860.00830.0133
0.01930.04590.05620.0419
0.00800.05520.03660.0186
0.03860.01760.04290.0133

DM(:,:,3,1) =
0.02310.04120.02210.0138
0.04120.03010.05450.0507
0.02030.05650.06530.0299
0.05350.01460.04220.0171

DM(:,:,4,1) =
0.04820.02610.01910.0128
0.02610.05570.06400.0617
0.01930.06450.06050.0196
0.06270.01830.06530.0309

DM(:,:,5,1) =
0.07980.03840.05950.0125
0.03840.08110.06650.1042
0.06450.06170.11900.0557
0.10640.01460.06450.0733

DM(:,:,1,2) =
0.15330.01680.11470.1207
0.02060.19800.07980.0758
0.13580.07680.11800.0151
0.06750.15430.20630.1109

DM(:,:,2,2) =
0.17490.02260.12800.1335

0.02610.20100.08180.0715
0.14130.07300.12950.0196
0.06400.15460.19650.1114

DM(:,:,3,2) =
0.15860.06830.16190.1860
0.07550.18750.06900.0489
0.16540.06580.17290.0402
0.03060.21030.17940.1742

DM(:,:,4,2) =
0.21030.01980.07980.0929
0.01780.23820.16970.1400
0.11120.13980.15330.0166
0.15110.08630.21610.1345

DM(:,:,5,2) =
0.13880.01860.07830.1019
0.02030.15660.08810.0580
0.10090.06370.12300.0261
0.06100.10310.13480.1107

DM(:,:,1,3) =
0.00790.00520.00220.0022
0.00510.00880.01170.0093
0.00100.01370.00730.0054
0.00630.00820.01060.0012

DM(:,:,2,3) =
0.00500.00460.00140.0010
0.00410.00590.00880.0049
0.00080.00950.00630.0023
0.00550.00470.00780.0052

DM(:,:,3,3) =
0.00570.00570.00250.0037
0.00570.00570.01790.0107
0.00180.01920.01200.0040
0.01120.00900.01530.0086

DM(:,:,4,3) =
0.01010.01300.00350.0029
0.01260.01190.02150.0118
0.00250.02230.01780.0048
0.01400.01000.01790.0018

DM(:,:,5,3) =
0.00650.01220.00250.0041
0.01150.00810.01770.0084
0.00200.01800.01470.0056
0.00910.01010.01210.0012

DM(:,:,1,4) =
0.03340.00850.03410.0284
0.01150.05400.02130.0256
0.01830.05220.05370.0339
0.03090.04620.03260.0612

DM(:,:,2,4) =
0.04420.00830.02260.0226
0.00800.05220.02840.0336
0.02560.02660.04640.0226
0.03390.02690.02790.0512

DM(:,:,3,4) =
0.14050.01080.08330.0791
0.01280.15810.06250.0808
0.06200.08930.13850.0557
0.07330.09490.08710.1496

DM(:,:,4,4) =
0.19750.01810.13400.1152
0.02360.24090.09640.1112
0.09360.14030.20600.0936
0.10920.13630.10060.2050

DM(:,:,5,4) =
0.03210.00450.02060.0206
0.00600.04170.02160.0178
0.02080.02030.03290.0173

0.01680.02580.02310.0309

C.1.3 Liver (Ns = 4)
DM(:,:,1,1) =
0.01220.00000.00490.0049
0.00000.01150.00540.0047
0.00580.00500.01110.0051
0.00570.00590.00550.0111

DM(:,:,2,1) =
0.01220.00000.00490.0049
0.00000.01150.00540.0047
0.00580.00500.01110.0051
0.00570.00590.00550.0111

DM(:,:,3,1) =
0.01220.00000.00490.0049
0.00000.01150.00540.0047
0.00580.00500.01110.0051
0.00570.00590.00550.0111

DM(:,:,4,1) =
0.02600.00010.01230.0127
0.00010.02510.01380.0122
0.01500.01200.02690.0128
0.01370.01410.01290.0265

DM(:,:,5,1) =
0.02780.00010.01290.0131
0.00010.02870.01570.0141
0.01570.01330.02870.0135
0.01480.01450.01390.0277

DM(:,:,1,2) =
1.0e-03 *
0.3774 -0.01960.20540.1662
 -0.02550.34700.20780.1436
0.18970.22690.38040.1372
0.19800.16220.16270.3867

DM(:,:,2,2) =
1.0e-03 *
0.4843 -0.00490.23230.1500
0.00690.42150.22110.1745
0.23280.17550.40240.1598
0.18430.24510.17500.3823

DM(:,:,3,2) =
1.0e-03 *
0.46910.04360.22740.1770
-0.02550.38870.21960.2152
0.19310.21180.39700.1917
0.17650.22500.17600.3814

DM(:,:,4,2) =
1.0e-03 *
0.52250.00390.21860.2765
-0.02890.56470.31810.2667
0.29170.24610.53480.2760
0.25050.27550.22650.5005

DM(:,:,5,2) =
1.0e-03 *
0.31570.00980.19170.1294
-0.02840.27300.16710.1127
0.17110.16910.32690.1049
0.14560.15830.14220.1426

DM(:,:,1,3) =
1.0e-03 *
0.12890.00350.07420.0718
0.00390.14950.08610.0686
0.08650.07160.15300.0703
0.08100.07760.15470.0774

DM(:,:,2,3) =
1.0e-03 *
0.21010.00350.09930.0974
0.00330.21390.12190.1032

0.12160.10510.22200.1025
0.11310.11540.11160.1990

DM(:,:,3,3) =
1.0e-03 *
0.34720.00370.17420.1623
0.00430.35430.20550.1675
0.20190.16480.35830.1649
0.18310.18750.18090.3280

DM(:,:,4,3) =
1.0e-03 *
0.30330.00660.15150.1373
0.00670.30510.17740.1484
0.17580.14750.31310.1422
0.16080.15840.15630.2741

DM(:,:,5,3) =
1.0e-03 *
0.52520.00850.25890.2430
0.00770.52280.29400.2505
0.29600.24920.52770.2460
0.27090.27140.26500.4705

DM(:,:,1,4) =
1.0e-04 *
0.55900.03800.29430.2218
0.04020.55750.34410.2193
0.30680.27450.53640.2127
0.29580.31430.28920.3881

DM(:,:,2,4) =
1.0e-04 *
0.34100.04080.18070.1449
0.04200.34290.21680.1537
0.19170.18980.33690.1609
0.19420.19830.18260.2641

DM(:,:,3,4) =
1.0e-04 *

0.34040.04170.17820.1496
0.04110.33690.20520.1531
0.19140.18450.33220.1581
0.19540.20050.18670.2663

DM(:,:,4,4) =
1.0e-04 *
0.27510.03760.15000.1321
0.03760.27420.16810.1343
0.16120.14560.27010.1340
0.15180.15590.15210.1528

DM(:,:,5,4) =
1.0e-04 *
0.23590.03730.13300.2372
0.03800.23210.14520.1183
0.14210.12140.22340.1120
0.12360.12890.12920.1860

Bibliography

[1] A. Jemal, A. Thomas, T. Murray, M. Thum, "Cancer Statistics, 2002," *CA Cancer J Clin*., Vol. 52, pp. 23-47; (2002).

[2] C. Morton, R. MacKie "Clinical accuracy of the diagnosis of cutaneous malignant melanoma", *British Journal of Dermatology*; Vol. 138, pp. 283-7; (1998).

[3] S. L. Jacques, J. C. Ramella-Roman, K. Lee. "Imaging skin pathology with polarized light," *J. Biomed. Optics,* Vol. 7, pp. 329-340, (2002).

[4] S. Gonzalez, M. Rajadhyaksha, R. R. Anderson, "Non-invasive (real-time) imaging of histologic margin of a proliferative skin lesion *in vivo*," *J. Invest. Derm*., Vol. 111(3), pp. 538-539; (1998).

[5] P. Anikijenko, L. T. Vo, E. R. Murr, J. Carrasco, W. J. McLaren, Q. Chen, S. G. Thomas, P. M. Delaney, and R. G. King "In vivo detection of small subsurface melanomas in athymic mice using noninvasive fiber optic confocal imaging," *J. Invest. Dermatol*., Vol. 117(6), pp. 1442-1448; (2001).

[6] K. J. Busam, K. Hester, C. Charles, D. L. Sachs, C. R. Antonescu, S. Gonzalez, A. C. Halpern "Detection of clinically amelanotic malignant melanoma and assessment of its margins by in vivo confocal scanning laser microscopy" *Arch Dermatol*., Vol. 137(7), pp. 923–929, (2001).

[7] J. F. de Boer, T. E. Milner, M. J. C. van Gemert, and J. S. Nelson, ''Two-dimensional birefringence imaging in biological tissue by polarization-sensitive optical coherence tomography,'' *Optics Letters*, Vol. 22, pp. 934–936; (1997).

[8] M. Rajadhyaksha, S. González, J. M. Zavislan, R. R. Anderson, R. H. Webb, "In vivo confocal scanning laser microscopy of human skin II: advances in instrumentation," *J. Invest. Dermatol*., Vol. 113, pp. 293-303; (1999).

[9] M. Rajadhyaksha, G. Menaker, T. Flotte, P. J. Dwyer, and S. Gonzalez "Confocal examination of nonmelanoma cancers in thick skin excisions to potentially guide Mohs micrographic surgery without frozen histopathology," *J. Invest. Dermatol.*, Vol. 117(5), pp. 1137–1143; (2001).

[10] S. Gammal, C. El Gammal, K. Kaspar, C. Pieck, P. Altmeyer, M. Vogt, H. Ermert, "Sonography of the skin at 100 MHz enables in vivo visualization of stratum corneum and viable epidermis in palmar skin and psoriatic plaques" *J Invest Dermatol*, Vol. 113 (5), pp. 821-829; (1998).

[11] H. Song, F. Wehrli, J. Ma, "In vivo MRI microscopy of human skin" *Magn Reason Med*, Vol. 37, pp. 185-191; (1997).

[12] P. Asawanonda, C. R. Taylor, "Wood's light in dermatology," *International Journal of Dermatology*, Vol. 38, No. 11, pp. 801; (1999).

[13] M. Moncrieff, S. Cotton, E. Claridge, P. Hall, "Spectrophotometric intracutaneous analysis: a new technique for imaging pigmented skin lesions," *British Journal of Dermatology*, Vol. 146, pp. 448–457;(2002).

[14] L. K. Rodrigues, S. P. Leong, B. M. Ljung, R. W. Sagebiel, N. Burnside, T. Hu, B. W. N. Miller, M. Kashani-Sabet, "Fine needle aspiration in the diagnosis of metastatic melanoma", *J. Am. Acad. Dermatol*., Vol. 46 (1) pp. 147-148; (2002).

[15] R. Wehner, "Polarized-light navigation by insects," *Scientific American*, Vol. 31(1), pp.106; (1976).

[16] H. LaFey, "The Vikings," *National Geographic*, Vol. 137, pp. 528, (1970)

[17] Bartholinus, Hafniae, (1670). Reprinted in ref 6.

[18] W. Shurcliff, S. Ballaed, *Polarized Light: Production and Use*, Harvard University Press, Cambridge; (1962).

[19] C. Brosseau, *Fundamentals of Polarized Light: A Statistical Optics Approach*, Wiley-Interscience Pubblication; (1999).

[20] S. Bianchi, A. Ferrara, C. Giovannardi, “Monte Carlos simulations of dusty spiral galaxies: extinction and polarization properties,” *American Astronomical Society*, Vol. 465 (1), pp. 137-144; (1996)

[21] G. W. Collins, P. F. Buerger, T. Gehrels, *Planets, stars, and nebulae studied with photopolarimetry* University of Arizona Press; (1974)

[22] G. W. Kattawar, C. N. Adams “Stokes vector calculations of the submarine light field in an atmosphere-ocean with scattering according to a Rayleigh phase matrix: effect of interface refractive index on radiance and polarization,” *Limnol. Oceanogr*., Vol. 34(8), pp. 1453-1472; (1989)

[23] G. W. Kattawar, G. N. Plass “Radiance and polarization of multiple scattered light from haze and clouds”, *Applied Optics*, Vol.7, No. 8, pp. 1519-1527; (1967)

[24] W. C. McCrone, L. B. McCrone, J. C. Delly, *Polarized light microscopy* McCrone Research Institute; (1987)

[25] M. M. Frocht, *Photoelasticity*, Vol 1., John Whiley & Sons, Inc.; (1946)

[26] W. S. Bickel, J. F. Davidson, D. R. Huffman, R. Kilkson, “Application of polarization effects in light scattering: a new biophysical tool,” *Proc Natl Acad Sci*; Vol. 73, pp. 486 – 490, (1976)

[27] Anderson RR. “Polarized light examination and photography of the skin”. *Arch Dermatol*; Vol. 127, pp. 1000–1005; (1991)

[28] J. M. Schmitt, A. H. Gandjbakhche, and R. F. Bonner, ‘‘Use of polarized light to discriminate short-path photons in a multiply scattering medium,’’ *Applied Optics,* Vol. 32, pp. 6535–6546; (1992)

[29] S. L. Jacques, L. H. Wang, D. V. Stephens, M. Ostermeyer, “Polarized light transmission through skin using video reflectometry: toward optical tomography of superficial tissue layers,” *Proc SPIE: Lasers in surgery: advanced characterization, therapeutics, and systems VI*. Vol. 2671, pp. 199–220, (1996)

[30] M. R. Ostermeyer, D. V. Stephens, L. Wang, and S. L. Jacques, ‘‘Nearfield polarization effects on light propagation in random media,’’ *OSA*

TOPS on Biomedical Optical Spectroscopy and Diagnostics, E. Sevick-Muraca and E. Benaron, Eds., Vol. 3, pp. 20–25, Optical Society of America, Washington, DC; (1996)

[31] A. H. Hielscher, J. R. Mourant, I. J. Bigio, "Influence of particle size and concentration on the diffuse backscattering of polarized light from tissue phantoms and biological cell suspensions," *Applied Optics*; Vol. 36, pp. 125–135; (1997)

[32] D. J. Maitland, J. T. Walsh Jr., "Quantitative measurements of linear birefringence during heating of native collagen," *Lasers Surg. Med*.; Vol. 30, pp. 310–318; (1997).

[33] V. Sankaran, J. T. Walsh Jr, "Birefringence measurement of rapid structural changes during collagen denaturation," *Photochem. Photobiol*.; Vol. 8, pp. 846–851; (1998).

[34] J. R. Mourant, A. H. Hielscher, J. P. Freyer, A. A. Eick, D. Shen, T. M. Johnson. "Scattering properties of biological cells" *OSA Trends in Optics NA Photonics: Biomedical Optical Spectrosopy Dicagnostic II*, Vol 22, E. Sevick-Mu raca and D. Benaro, eds., Optical Society of America, Washington DC, pp. 11-15; (1998).

[35] M. Canpolat, J. R. Mourant, "Particle size analysis of turbid media with a single optical fiber in contact with the medium to deliver and detect white light," *Applied Optics-OT*, Vol. 40, No. 22, pp. 3792–3799; (2001).

[36] G. Jarry, E. Steimer, V. Damaschini, M. Epifanie, M. Jurczak, R. Kaiser, "Coherence and polarization of light propagating through scattering media and biological tissues, "*Applied Optics*; Vol. 37, pp. 7357–7367; (1998).

[37] S. L. Jacques, R. J. Roman, K. Lee, "Imaging superficial tissues with polarized light," *Lasers Surg. Med*. Vol. 26, pp. 119–129; (2000).

[38] V. Backman, R. Gurjar, K. Badizadegan, R. Dasari, I. Itzkan, L. T. Perelman, M. S. Feld, "Polarized light scattering spectroscopy for quantitative

measurement of epithelial cellular structures *in situ*," *IEEE JSTQE*, Vol. 5, pp. 1019-1027; (1999).

[39] L. T. Perelman, V. Backman, M. Wallace, G. Zonios, R. Manoharan, A. Nusrat, S. Shields, M. Seiler, C. Lima, T. Hamano, I. Itzkan, J. Van Dam, J. M. Crawford, M. S. Feld, "Observation of periodic fine structure in reflectance from biological tissue: a new technique for measuring nuclear size distribution," *Phys. Rev. Lett*., Vol. 80, pp. 627–630; (1998).

[40] K. Sokolov, R. Drezek, K. Gossage, and R. Richards-Kortum, ''Reflectance spectroscopy with polarized light: is it sensitive to cellular and nuclear morphology,'' *Opt. Express,* Vol. 5, pp. 302–317; (1999).

[41] A. Ambirajan, D.C. Look, "A backward Monte Carlo study of the multiple scattering of a polarized laser beam," *J.Quant.Spectrosc. Radiat. Transfer,* Vol. 58, No. 2, pp.171-192; (1997).

[42] S. Bartel, A. Hielsher, "Monte Carlo simulations of the diffuse backscattering Mueller matrix for highly scattering media," *Applied Optics*, Vol. 39, No. 10; (2000).

[43] Nelder, J. A. and R. Mead, "A Simplex method for function minimization", Computer Journal , Vol. 7, p. 308-313.

[44] B. D. Cameron, M. J. Rakovic, M. Mehrubeoglu, G. Kattawar, S. Rastegar, L.-H. Wang, G. L. Cote, "Measurement and calculation of the two-dimensional backscattering Mueller matrix of a turbid medium," *Optics Letters*, Vol. 23, pp. 485-487; (1998).

[45] M. J. Rakovic, G. W. Kattawar, M. Mehrubeoglu, B. D. Cameron, L.-H. Wang, S. Rastegar, G. L. Cote, "Light backscattering polarization patterns from turbid media: theory and experiment," *Applied Optics*, Vol. 38, pp. 3399-3408; (1999).

[46] A. S. Martinez, R. Maynard, "Polarization statistics in multiple scattering of light: a Monte Carlo approach," in *Localization and Propagation of classical waves in random and periodic structures*, Plenum Publishing Corporation New York

[47] X. Wang, L. V. Wang, "Propagation of polarized light in birefringent turbid media: time-resolved simulations," *Optics Express*, Vol. 9. No. 5, 254-259; (2001).

[48] Max Born and Emil Wolf, Principles of Optics, The Macmillan Company, (1964)

[49] D Clarke, J. F. Granger, *Polarized Light and Optical Measurements*, Pergemon Press; (1971).

[50] Serge Huard, *Polarization of Light*, Whiely & Sons; (1997).

[51] G. G. Stokes, *Trans. Cambridge. Phil. Soc*. 9, 309; (1852).

[52] S. L. Jacques, J.C. Ramella-Roman, and K. Lee, "Imaging skin pathology with polarized light," *Journal of Biomedical Optics.*, Vol. 7, No. 3, pp. 329-340; (2002).

[53] M.H. Smith, P. Burke, A. Lompado, E. Tanner, L.W. Hillman "Mueller matrix imaging polarimetry in dermatology" *Proc. of SPIE* Vol. 3911; (2000).

[54] J. A. Muccini, N. Kollias, S. B. Phillips, R. R. Anderson, A. J. Sober, M. J. Stiller, L. A. Drake. "Polarized light photography in the evaluation of photoaging," *J. Am. Acad. Dermatol.*; Vol. 33, pp. 765-769; (1995).

[55] G. Argenziano, G. Fabbrocini, P. Carli, V. De Giorgi, E. Sammarco, M. Delfino "Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions," *Arch. Dermatol.*; Vol. 134, pp. 1563-1570; (1998)

[56] R. Langley, M. Rajadhyaksha, P. Dwyer, A. Sober, T. Flotte, R.R. Anderson. "Confocal scanning laser microscopy of benign and malignant melanocytic skin lesions *in vivo*," *Journal of the American Academy of Dermatology*; Vol. 45, pp. 365-376; (2001)

[57] S. González, Sackstein, M. Rajadhyaksha, R. R. Anderson, "Real-time evidence of in vivo leukocyte trafficking in human skin by reflectance confocal microscopy," *Journal of Investigative Dermatology*, Letter to the Editor, Vol. 117, pp. 384-386; (2001)

[58] http://developer.apple.com

[59] G. Towner, *Discovering Quicktime: An introduction for windows and macintosh programmer*, Morgan Kauffman, San Francisco; (1999)

[60] P. R. Bargo, S. A. Prahl, and S. L. Jacques, "Optical properties effects upon the collection efficiency of optical fibers in different probe configurations," *IEEE-Journal of Select Topics in Quantum Electronics*, Vol. 9, No. 2; (2003).

[61] J. C. Ramella-Roman, S. A. Prahl, S. L. Jacques, "Three polarized light Monte Carlo programs, a performance comparison" (In preparation).

[62] G. W. Kattawar and G.N. Plass, "Radiance and Polarization of Multiple Scattered Light from Haze and Clouds," *Applied Optics*, Vol. 7, No. 8; (1968).

[63] S. Chandrasekhar, *Radiative Transfer*, Oxford Clarendon Press; (1950).

[64] S. Chandrasekhar, and D. Elbert, "The Illumination and polarization of the sunlight sky on Rayleigh scattering," *Trans. Am. Phil. Soc.*, New Series, Vol. 44, No. 6; (1956).

[65] K. F. Evans and G. L. Stephens, "A new polarized atmospheric radiative transfer model" *J.Quant.Spetrcosc. Radiat Transfer.*, Vol. 46, No. 5, pp. 413-423; (1991).

[66] S. A. Prahl, M. J. C. van Gemert, A. J. Welch, "Determining the optical properties of turbid media by using the adding-doubling method," *Applied Optics*, Vol. 32, pp. 559-568, (1993).

[67] G. W. Kattawar G. N. Plass, "Degree and direction of polarization of multiple scattered light. 1: Homogeneus cloud layers," *Applied Optics*, Vol. 11, No. 12 pp. 2851-2865; (1972).

[68] P. Bruscaglione G. Zaccanti W. Qingnong "Transmission of a pulsed polarized light beam through thick turbid media: numerical results," *Applied Optics*, Vol. 32, No. 30, pp. 6142-6150; (1993).

[69] S. Bianchi *Estinzione e polarizzazione della radiazione nelle galassie a spirale*, Tesi di Laura (in Italian); (1994).

[70] A. S. Martinez and R. Maynard "Polarization Statistics in Multiple Scattering of light: a Monte Carlo approach" in *Localization and Propagation of*

classical waves in random and periodic structures Plenum Publishing Corporation New York

[71] A. S. Martinez *Statistique de polarization et effet Faraday en diffusion multiple de la lumiere* Ph.D. Thesis (in French and English); (1984).

[72] N. Metropolis, S. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association*, Vol. 44, pp. 335-341; (1949)

[73] S. A. Prahl, M. Keijzer, S. L. Jacques, A. J. Welch, "A Monte Carlo model of light propagation in tissue," *In Dosimetry of Laser Radiation in Medicine and Biology , G. Mueller, D. Sliney, Eds., SPIE Series* Vol. IS 5, pp. 102-111, (1989).

[74] L-H Wang, S. L. Jacques, L-Q Zheng, "MCML - Monte Carlo modeling of photon transport in multi-layered tissues" *Computer Methods and Programs in Biomedicine*, Vol. 47, pp. 131-146; (1995).

[75] A. N. Witt, "Multiple scattering in reflection nebulae I: A Monte Carlo approach," *Astophys. J. Suppl. Ser.*, Vol. 35, pp. 1-6; (1977).

[76] C. Bohren and D.R. Huffman, *Absorption and scattering of light by small particles*, Wiley Science paperback Series; (1998).

[77] http://omlc.ogi.edu/software/mie/index.html

[78] W. J. Wiscombe, "Improved Mie scattering algorithms", *Applied Optics*, Vol. 19 (9), pp. 1505-1509; (1980).

[79] J. W. Hovenier, "Symmetry Relationships for Scattering of Polarized Light in a Slab of Randomly Oriented Particles" *Journal of the Atmospheric Sciences*, Vol. 26, pp 488-499; (1968).

[80] J. J. Craig *Introduction to robotics. Mechanics and controls* Addison-Weseley Pubblishing Company; (1986).

[81] D. Benoit, D. Clary "Quaternion formulation of diffusion quantum Monte Carlo for the rotation of rigid molecules in clusters" *Journal of chemical physics*, Vol. 113, No. 12, pp. 5193-5202; (2000).

[82] M. Richartz., Y. Hsu, "Analysis of elliptical polarization",. *J. Opt. Soc. Am.*, Vol. 39, 136-157; (1949)

[83] K. I. Hopcraft, P. C. Y. Chang, J. G. Walker, E. Jakeman, *Properties of Polarized light-beam multiply scattered by Ryaleigh medium*, F. Moreno and F. Gonzales (Eds.): Lectures 1998, LNP 534, pp. 135-158, Springer-Verlag; (2000).

[84] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C, the art of Scientific Computing*, Cambridge University Press; (1992).

[85] G. Yao, L-H. Wang, "Two-dimensional depth-resolved Muller matrix characterization of biological tissue by optical coherence tomography," *Optics Letters*, Vol. 24, pp. 537-539; (1999).

[86] G. Burg, *Atlas of cancer of the skin*, Churchill Livingstone; (2000)

[87] V. Sankaran, *Polarized light scattering by tissue and tissue phantoms*, Ph.D Thesis; (1999).

[88] R. C. Studinski and I. A. Vitkin, "Methodology for examining polarized light interactions with tissues and tissue-like media in the exact backscattering direction," *J. Biomedical Optics*, Vol. 5, pp. 330-337; (2000).

[89] J. C. Ramella-Roman and S. L. Jacques, 'Mueller matrix description of collimated light transmission through liver, muscle and skin.' *SPIE proceedings* Vol. 3915; (2000).

[90] S. L. Jacques, A. Moody, J. C. Ramella Roman, "Characterizing microscopic domains of birefringence in thin tissue sections," *Proceedings of SPIE, Optical Technologies to Solve Problems in Tissue Engineering*, Vol. 4257B; (2001).

[91] W. F. Cheong, S. A. Prahl and A. J. Welch, "A review of the optical properties of biological tissue," *IEEE J. Quantum Electron.*, Vol. 26, pp. 2166-2185; (1990).

[92] M. Kerker *The scattering of light and other electromagnetic radiation*, Academic Press; (1969).

[93] D. Bicout, C. Brosseau, A. S. Martinez, J. M. Schmitt, "Depolarization of multiply scattered waves by spherical diffusers: influence of size parameter," *Phys. Rev.*, E 49, pp. 1767-1770; (1994).

[94] J. Cariou, B. Le Jeune, J. Lotrian, and Y. Guern , "Polarization effects of seawater and underwater targets", *Applied Optics*, Vol. 29, pp. 1689-1695; (1990).

[95] J. G. Walker, P. C. Y. Chang, K. I. Hopcraft, "Visibility depth improvement in active polarization imaging in scattering media," *Applied Optics-LP*, Vol. 39, No. 27, pp. 4933; (2000).

[96] P. C. Y. Chang, J. C. Flitton, K. I. Hopcraft, E. Jakeman, D. L. Jordan, J. G. Walker, "Improving visibility depth in passive underwater imaging by use of polarization," *Applied Optics-LP*, Vol. 42, No. 15, pp. 2794; (2003).

[97] E. Collett, *Polarized Light*, Marcell Dekker Inc.; (1993).

[98] P.S. Hauge, "Mueller matrix ellipsometry with imperfect compensators", *J. Opt. Soc. Am*., Vol. 68, No. 11; (1978)

[99] S. F. Nee, "Error analysis for Mueller Matrix measurements" *J. Opt. Soc. Am*., Vol. 20, No.8. (2003).

[100] A. Stanworth, E. J. Naylor, "Polarized light studies of the cornea" *J Exp Biol*., Vol. 30, pp.160-163; (1953).

[101] K. Sokolov, R. Drezek, K. Gossage, R. Richards-Kortum, "Reflectance spectroscopy with polarized light: is it sensitive to cellular and nuclear morphology," *Optics Express*, Vol. 5 No. 13 pp. 302-317; (1999).

[102] J. D. Khaydarov, B. Ovryn, "Measurement of three-dimensional velocity profiles in a thin channel flow using forward scattering particle image velocimetry (FSPIV)," *Proceedings ASME Fluids Engineering Division Summer Meeting*, pp. 4403–408; (1996).

[103] B. Ovryn, T. Wright, J. D. Khaydarov, "Measurement of three-dimensional velocity profiles using forward scattering particle image velocimetry (FSPIV) and neural net pattern recognition, " *Proceedings SPIE* Vol. 2546 pp. 112–124; (1995).

[104] B. Kachar "Asymmetric illumination contrast: a method of image formation for video light microscopy," *Science*, Vol. 227, pp. 766–768, (1984).

[105] F. Bretshneider & P. F. Teunis, “Reduced-carrier single-sideband microscopy: a powerful method for the observation of transparent microscopical objects,” *Journal of Microscopy*, Vol. 175, No. 2, pp. 121–134; (1994).

[106] G. Greenberg, “Direct 3-D imaging using a multiple oblique microscope,” *Scanning* Vol. 16, No. 4 pp. 248–249, (1994).

[107] G. Karp *Cell and Molecular Biology, concepts and experiments*, John Wiley & Sons, Inc.; (1996).

Bibliographical Note

Jessica C. Ramella-Roman was born in Pistoia, Italy. She attended the Polytechnic of Milan from 1986 to 1988 and was a student of the University of Pavia, Italy from 1989 to 1993. In 1990 she attended classes in Electrical Engineering at Strathclyde University of Glasgow, UK supported by the Erasmus sponsorship of the EU. She was a visiting student at the Laboratoire de Physique des Lasers at the University Paris XIII, Paris France from 1992-1993. She completed her studies at the University of Pavia, Italy in 1993 obtaining a *Laurea* in Electrical Engineering with a minor in Bioengineering. From 1993 to 1998 she worked in the semiconductor field as an engineer and a manager. In 1999 she entered the Oregon Graduate Institute to pursue a PhD degree with Professor Steve Jacques. In 2003 she was the winner of the Paul Clayton Student achievement award. Jessica is currently a postdoctoral fellow at the Applied physics laboratory at the Johns Hopkins University, USA.