A FLUID MECHANICS APPROACH TO UNDERSTANDING FIBRILLAR STRUCTURAL ORGANIZATION IN 2D AND 3D

By

Adriana C. Salazar Coariti

B.S. Mechanical Engineering, University of Illinois at Urbana-Champaign, 2018

Thesis

Submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Biomedical Engineering at Brown University

PROVIDENCE, RHODE ISLAND

October 2025

This dissertation by ADRIANA C. SALAZAR COARITI is accepted in its present form by the Department of BIOMEDICAL ENGINEERING as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date 8/27/25

KIMANI C. TOUSSAINT JR., Advisor

Recommended to the Graduate Council

Date 7/22/25

JEFFREY MORGAN, Reader

Date 7/22/2025

EDITH MATHIOWITZ, Reader

Date 7-22-2025

ROBERTO ZENIT, Reader

Approved by the Graduate Council

Date _____

THOMAS A. LEWIS, Dean of the Graduate School

ADRIANA C. SALAZAR COARITI

+1 (206) 474 7037 ⋄ adriana_carola _salazar@brown.edu
Brown University School of Engineering, 184 Hope St, Providence, RI 02912, United States of America

EDUCATION

Brown University 2020-2025
Ph.D. in Biomedical Engineering *Providence, RI, USA*
Advisor: Prof. Kimani C. Toussaint Jr. Research area: Quantification of complex (2D/3D) collagen structures and label-free imaging (Second harmonic generation & Polarization)

University of Illinois at Urbana-Champaign 2015-2018
B.S. in Mechanical Engineering *Urbana-Champaign, IL, USA*

PUBLICATIONS

1. **Adriana C. Salazar Coariti**, Nicholas Gaitanis, Caitlin Hopkins, Jeffrey Morgan, Kimani C. Toussaint Jr. "A fluid mechanics approach to quantify 3D collagen organization." *(Submitted)*

2. Jourdan K. Ewoldt, Samuel J. DePalma, Maggie E. Jewett, M Çağatay Karakan, Yih-Mei Lin, Paria Mir Hashemian, Xining Gao, Lihua Lou, Micheal A. McLellan, Jonathan Tabares, Marshall Ma, **Adriana C. Salazar Coariti**, Jin He, Kimani C Toussaint Jr, Thomas G. Bifano, Sharan Ramaswamy, Alice E. White, Arvind Agarwal, Emma Lejeune, Brendon M. Baker, Christopher S. Chen. "Induced pluripotent stem cell-derived cardiomyocyte in vitro models: benchmarking progress and ongoing challenges." *Nature Methods* (2024): 1-17.

3. Ian Gonsher, **Adriana Salazar**, Shrey Mehta, Samantha Shulman, Nicholas Gaitanis, Arshiya Khosla, Denise Danielle Tamesis and Jillian Sun. "The Smart Cane Project: Integrating Screen Interfaces and Physiological Sensors into Mobility Devices." *AHFE International* (2023)

4. **Adriana C. Salazar Coariti**, Maurice S Fabien, Johnny Guzman, Jeffrey A McGuire, Raffaella De Vita, Kimani C Toussaint Jr. "Fluid mechanics approach to analyzing collagen fiber organization." *Journal of Biomedical Optics* 27, no. 1 (2022): 016503-016503.
— **Cover Feature**, January 2022 Issue

5. Yuhang Yang, Varun A Kelkar, Hemangg S Rajput, **Adriana C. Salazar Coariti**, Kimani C Toussaint Jr, Chenhui Shao. "Machine-learning-enabled geometric compliance improvement in two-photon lithography without hardware modifications." *Journal of Manufacturing Processes* 76 (2022): 841-849.

6. Jeffrey A McGuire, Jose L Monclova, **Adriana C. Salazar Coariti**, Caleb A Stine, Kimani C Toussaint Jr, Jennifer M Munson, David A Dillard, Raffaella De Vita. "Tear propagation in vaginal tissue under inflation." *Acta biomaterialia* 127 (2021): 193-204.

7. Mohammad M Kabir, Hemangg S Rajput, Varun A Kelkar, **Adriana C. Salazar Coariti**, Kimani C Toussaint Jr. "Demonstration of flat-top beam illumination in widefield multiphoton microscopy." *Journal of Biomedical Optics* 25, no. 1 (2020): 014503-014503.

8. **Adriana C. Salazar Coariti**, Maurice Fabien, Johnny Guzman, Kimani C Toussaint. "Quantitative second-harmonic generation imaging analysis based on fluid-dynamics measures." *Frontiers in Optics*, pp. FW5E-1. Optica Publishing Group, 2020.

AWARDS

- *First Prize*, Engineering Research Center in Cellular Metamaterials (CELL-MET) Pitch Perfect competition
- *Fellowship*, Engineering Research Center in Cellular Metamaterials (CELL-MET) Science Communication
- *Award*, James & Loretta Bayne Mechanical Engineering Outstanding Senior Design

ACADEMIC SERVICES

- Reviewer for: *Journal of Biomedical Optics*
- Poster Judge for: *2025 SACNAS New England Chapters Community Gathering*

TECHNICAL SKILLS

Research	Nanoscribe, ImageJ, Laser beam alignment, LabVIEW, OriginPro, Two-photon lithography and Adobe Illustrator
Microscopy	Brightfield, Phase, DIC, Polarization, Confocal, Multiphoton, SHG Light-sheet, SEM and TEM
Programming	MATLAB, Mathematica, NumPy, Python, Html, JavaScript, Java Eclipse
Computer Aided Design	Creo Parametric 3.0, Solid Works and AutoCAD
Databases	MySQL and PostgreSQL
Geographical Analysis	QGis and POSTGIS
Languages	Spanish and English

EXPERIENCE

Brown University September 2020 - Present
Research Assistant *Providence, RI*

· Developed algorithms to quantify 2D and 3D fibrillar collagen fiber organization using finite difference methods and fluid mechanics-inspired analysis.
· Designed visual research representations featured on the cover of Journal of Biomedical Optics (Vol. 27, No. 1) and exhibited at the Warren Alpert Medical School.
· Engineered and refined multiple versions of a polarization microscope for label-free imaging of engineered cardiac tissue.
· Collaborated on studies investigating morphological changes in type I collagen during fibrosis and in mechanically stressed tissue models.
· Developed and implemented experiments to advance spatiotemporal light-sheet imaging techniques.

Brown University August 2019 - September 2020
Lab Manager *Providence, RI*

· Coordinated scientific collaborations to advance interdisciplinary research initiatives.
· Assessed laboratory needs and recommended equipment to improve efficiency in biological sample preparation workflows
· Supervised day-to-day laboratory operations, including inventory management, ordering supplies, and equipment maintenance.
· Trained graduate students in lab techniques, safety protocols, and data recording procedures.
· Ensured compliance with institutional biosafety and chemical hygiene standards.
· Assisted in writing the Biological Research Authorization (BRA) and follow protocols, as applicable.

National Center for Supercomputing Applications February 2019 - August 2019
Research Scholar *Urbana-Champaign, IL*

- Evaluated user interfaces of nanomanufacturing simulation tools to improve usability and engagement.
- Compiled and documented structured feedback for tool developers across the nanoBIO platform.
- Developed onboarding pathways and guidelines for new software tool developers and collaborators.
- Designed a framework to help developers systematically document their simulation tools.
- Analyzed and assessed front-end and back-end software testing methods to ensure tool reproducibility and reliability.
- Led the conceptualization of an educational video game aimed at introducing nanotechnology concepts to children.

Photonics Research of Bio/Nano Environments Laboratory December 2018 - August 2019
Research Assistant *Urbana-Champaign, IL*

- Modified photo polymer formulations originally used in stereolithography for compatibility with two-photon lithography and analyzed curing rates as well as efficiency.
- Constructed and aligned custom optical setups for two-photon-lithography experiments.
- Explored image-processing techniques to stitch and analyze SHG microscopic images using Python.
- Processed microscopy data to assess fiber orientation and structural features at the microscale using MATLAB.

Center for Innovation in Teaching & Learning September 2016 - October 2017
IT Support Specialist – Education Sector (Part-Time) *Urbana-Champaign, IL*

- Contributed to the development of web-based educational tools using HTML and CSS.
- Developed and maintained internal tools using Python and JavaScript to automate help desk workflows.
- Retrieved and configured legacy software by setting up virtual machines, enabling compatibility with modern operating systems.

CityHook & Indigo July 2016 - August 2016
Intern *Dublin, Ireland*

- Analyzed geographic and transport datasets to identify routing patterns and optimize geospatial routing strategies.
- Applied mathematical modeling techniques to interpret and structure geospatial information.
- Developed and optimized algorithms to improve runtime performance in geospatial data analysis.

PRESENTATIONS

Invited Talks

1. **Brown Institute of Biology, Engineering and Medicine Spring Retreat** May 2025
   Fluid-Inspired Fiber Analysis (FIFA) of 3D Collagen Architecture.
2. **Opening Ewgis Conference** July 2022
   Writing about engineering and math.

Oral Presentations

3. **Mechanobiology Methods: From Molecules to Cells** June 2025
   Lightening Talk: Fluid-Inspired Fiber Analysis (FIFA)
4. **Tissue Repair and Regeneration Gordon Research Seminar** June 2025
   Fluid-Inspired Fiber Analysis (FIFA) of 3D Collagen Architecture.

5. **2025 SACNAS New England Chapter Community Gathering** May 2025
   Fluid-Inspired Fiber Analysis (FIFA) of 3D Collagen Architecture.
6. **2022 SACNAS New England Chapter Community Gathering** June 2022
   Fluid mechanics approach to analyzing collagen fiber organization
7. **Frontiers in Optics (FiO)** September 2020
   Quantitative second-harmonic generation imaging analysis based on fluid-dynamics measures.

Poster Presentations

8. **Tissue Repair and Regeneration Gordon Research Conference** June 2025
   Fluid-Inspired Fiber Analysis (FIFA) of 3D Collagen Architecture.
9. **International Conference on Mechanics of Biomaterials and Tissues** December 2023
   A Fluid Mechanics Approach To Quantify 3D Collagen Organization.
10. **Optics and Photonics in Medicine and Biology Gordon Conference** July 2022
    Quantification of non-centrosymmetric structures.

SCIENCE POPULARIZATION

1. **ICERM Girls Get Math** August 2022
   Ballistic Flight and Image Processing
   Website: Brown University program makes math fun, and lucrative
   Website: Girls Get Math program helps high school girls build confidence in math and science
2. **Worldwide Youth in Science and Engineering camp** August 2019
   Bubbles to Teach Fluids Principles
   Website: MechSE's Feng Uses Bubbles to Teach Fluids Principles, Pique Students' Interest in STEM
   Website: Adriana Carola Salazar Coariti Passionate About STEM Outreach, Cooking Up a Multidisciplinary Masterpiece
3. **Yankee Ridge Elementary, Junior Scientist Day Presenter** April 2019
   Introduction to nanotechnology and fabrication of gold nanoparticles.

LEADERSHIP AND VOLUNTEERING

Rhode Island Free Clinic November 2023 - Present
Medical Interpreter *Providence, RI*

- Provided accurate and confidential interpretation for patients and providers across specialties including endocrinology, cardiology, primary care, neurology, and physical therapy.
- Facilitated clear communication between patients and healthcare teams, helping overcome language and cultural barriers to improve quality of care.

Brown University Optics Society (BRUNOS) June 2021 - October 2023
Founder and President *Providence, RI*

- Established the BRUNOS's mission, structure, and bylaws to align with institutional policies and student interests.
- Recruited and built a membership through outreach, events, and consistent participation.
- Secured official recognition as a campus organization and successfully applied for OPTICA and SPIE student chapter status to fund club initiatives.
- Led executive board meetings and general body meetings.

- Organized campus-wide events, workshops, and initiatives that highlighted the interdisciplinary nature of optics and photonics.
- Managed communications, including social media, email campaigns, and collaborations with other student organizations.
- Oversaw budgeting, funding proposals, and relationships with faculty advisors and university departments.
- Facilitated and hosted research meetings to promote scientific exchange and interdisciplinary collaboration among students and faculty.

National Center for Supercomputing Applications August 2021 - August 2023
Project leader *Urbana-Champaign, IL*

- Led the planning and execution of a game designed to teach middle school students fundamental concepts in nanoscience.
- Mentored and supervised two students throughout the game development process, providing guidance on design and implementation.
- Conducted research on educational game design principles and identified technical and educational resources to support the project.

HOPE Worldwide Campus Club June 2018 - May 2019
Leader *Urbana-Champaign, IL*

- Researched and established partnerships with local organizations to coordinate volunteer opportunities in the Champaign area.
- Organized and led volunteer teams for community service events, including One Winter Night and Boneyard Creek Community Day.

Sutton House August 2016 - August 2017
President *Urbana-Champaign, IL*

- Led and facilitated weekly house and board meetings, ensuring productive communication and team alignment.
- Identified community challenges and implemented solutions to foster a supportive and inclusive environment for 12 members.
- Organized activities designed to recognize individual strengths and build a collaborative and empowering house culture.

Illini Solar Car August 2015 - October 2015
Mechanical Engineering and Solar System subteam member *Urbana-Champaign, IL*

- Evaluated and selected optimal fasteners for composite materials to enhance durability and structural integrity of solar car panels
- Executed solar array encapsulation processes to ensure operational readiness and environmental protection of photovoltaic systems for Illini Solar Car

TEACHING AND MENTORSHIP

Engineering Biomedical Systems [Instructor] 2024.07
Iterate week long course based on feedback from last year.

Engineering Biomedical Systems [Instructor] 2023.07
Designed and taught a week-long course for 15 high school students, introducing foundational concepts through interactive lessons and hands-on activities.

Designing Humanity Centered Technology: Smart Cane [Mentor] 2023.02 - 2023.05
Taught 15 students micro-controller principles and app development

Optical Microscopy: Fundamentals and Applications [Teacher Assistant] 2020.08 - 2020.12
Designed and taught graduate-level light microscopy laboratory course for 16 students

ACADEMIC REFERENCES

Kimani C. Toussaint Jr. PhD supervisor
Brown University School of Engineering kimani_toussaint@brown.edu

Jeffrey Morgan Collaborator and thesis advisor
Brown Institute of Biology and Medicine jeffrey_morgan@brown.edu

Jonghwan Lee Optics Student Chapter Advisor
Brown Institute of Biology and Medicine jonghwan_lee@brown.edu

Edith Mathiowitz Thesis advisor
Brown Institute of Biology and Medicine edith_mathiowitz@brown.edu

Roberto Zenit Thesis advisor
Brown School of Engineering roberto_zenit@brown.edu

Rafaella De Vita Collaborator
Virginia Tech devita@vt.edu

Acknowledgments

This work would not have been possible without the support, mentorship, and encouragement of many people.

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Kimani C. Toussaint Jr., for his unwavering support, thoughtful guidance, critical insights, and the many inspiring scientific discussions we shared. I am especially thankful for the freedom to explore my own ideas, and for his support of my career interests, particularly in mechanobiology, by encouraging me to attend conferences aligned with my passions. His mentorship has shaped both my scientific thinking and personal growth, and for that, I am truly grateful.

I also extend my sincere thanks to the members of my thesis committee: Dr. Jeffrey Morgan, Dr. Edith Mathiowitz , and Dr. Roberto Zenit for their helpful and timely feedback, support, mentorship, encouragement, and the thought-provoking discussions that strengthened this work.

To my PROBE lab mates and collaborators, thank you! I've truly appreciated your support, our laser alignment sessions, and the valuable feedback during lab meetings. These contributions have significantly shaped and strengthened my research. I'm especially grateful to my collaborators, whose need for robust quantification tools to capture the complexities of collagen architecture directly inspired and motivated this work.

I am incredibly grateful to the Institute of Biology, Medicine, and Engineering for

providing an enriching academic environment and the resources necessary to complete this work. I would also like to acknowledge the support from CELL-MET, which made this research possible and served as a valuable hub for learning about cardiac tissue engineering. CELL-MET has offered opportunities to strengthen my academic skills by learning from world experts and has provided key resources for my professional development.

To my friends and family, thank you for your love, patience, and unwavering belief in me throughout this long journey. I have been blessed with friends from multiple departments at Brown who have challenged my thinking both personally and professionally. I am deeply grateful for the scientific discussions, professional encouragement, laughter, kindness, and support during early mornings and late evenings.

My family has been my rock, nurturing my scientific curiosity since childhood. I am especially grateful to my grandmothers, whose courage to challenge norms in a patriarchal society has inspired my scientific journey. My mom, Monica Coariti de Salazar, has been the muse behind all of my work, while my dad, Peter Salazar Lopez, sparked my creativity from an early age. My sister, Claudia Salazar., has been a constant source of love and support through the most stressful moments, and my sister, Monica Camila Salazar., has kept me grounded and filled my days with laughter.

To Nicholas Leon and Coppernicus Cooper, thank you for always being by my side as the best companions through late nights and early mornings, and for your endless patience during the writing of this thesis.

Second to last, to everyone who has been part of this journey in big or small ways, muchísimas gracias! I feel deeply blessed for your presence in my life, and I have grown and learned so much from sharing life with you.

Finally, I give all thanks and glory to God, whose grace, wisdom, and strength have sustained me throughout this journey. His light has guided my path, and His creation has continually inspired my curiosity and wonder.

Contents

Acknowledgments **x**

1 Background **3**

1.1 Extracellular Matrix 3

1.2 Type I Collagen 5

1.3 Imaging 10

1.4 Fiber Quantification 12

2 Fluid Mechanics Approach to Analyzing 2D Collagen Fiber Organization **23**

2.1 Introduction 23

2.2 Materials and methods 25

2.3 Results and Discussions 28

2.4 Conclusion 33

3 Fluid-Inspired Fiber Analysis of 3D Collagen Architecture **37**

3.1 Introduction 37

3.2 Materials and methods 40

3.2.1 Engineered Ring-Shaped Tissues Preparation 40

3.2.2 Image Acquisition 41

3.2.3 Fiber quantification 41

3.3 Results 45

3.3.1 Analysis of Calibration Data Using FIFA algorithm 45
3.3.2 Fiber architecture changes in ring tissues over three developmental stages 48
3.3.3 TGF-β1 influence on ring- tissue formation across three developmental stages 51
3.4 Discussions 54
3.5 Conclusion 57

4 Conclusion and Future Work **62**
4.1 Conclusion 62
4.2 Future Work 62
4.2.1 Applications 63
4.2.2 Tool Development 63
4.2.3 Simulations 64

List of Figures

2.1 Diagram outlining the algorithm for calculating τ, U, and E 27

2.2 Representative synthetic images showing (a)(i) uniformly aligned fibers and (b)(i) wavy fibers, along with (ii) their corresponding vector field (VF) maps, (iii) the resulting streamlines, and (iv) histograms of the computed tortuosity values. 28

2.3 (a) Representative SHG image of bovine tendon, (b) corresponding streamline map derived from the vector field, and (c) distribution of tortuosity (τ) values. Scale bars correspond to 25µm. 29

2.4 Representative SHG images of cortical bone showing (a)(i) regions with uniformly aligned fibers and (b)(i) areas containing Haversian canals. (ii) Corresponding streamline maps and (iii) tortuosity (τ) distributions are shown for each case. Scale bars indicate 200µm. 30

2.5 Representative SHG images of rat vaginal tissue near the (a)(i) cervix and (b)(i) introitus. (ii) Corresponding streamline maps and (iii) distributions of tortuosity (τ) values are shown for each region. Scale bars denote 10µm. 31

3.1 Flow diagram of the algorithm used to convert image data to a 3D VF and the accompanying calculated fluid mechanics metrics 44

3.2 (A) Representative rendered straight and undulated phantom fiber volumes with overlaid 3D VFs. (B) Bivariate tiled histogram of in-plane and out-of-plane orientations. (C) Polar plots indicating orientation frequency of θ and ϕ angles throughout the volume 46

3.3 Calculated fluid-inspired metrics: (A) Streamlines, (B) (E_ρ) (light green) and (U_ρ) (dark green), and (C) Tortuosity values for phantom straight and undulated fibers 47

3.4 (A) Representative SHG images showing engineered 3D ring-shaped tissues over 21 days. (B) Volumetric SHG rendered images overlaid with computed 3D VF of collagen fiber orientations. (C) Bivariate tiled histogram of in-plane and out-of-plane orientations. (D) Polar plots indicating orientation frequency of θ and ϕ angles throughout the volume 49

3.5 Calculated fluid-inspired metrics over 21 days of ring tissue development: (A) Streamlines, (B) (E_ρ) (light green) and (U_ρ) (dark green), and (C) Tortuosity values highlighting architectural changes 50

3.6 (A) Representative SHG images showing the developmental effects of TGF-β1 on ring tissues over 21 days. (B) Collagen I fiber orientation VF from 3D rendered SHG images. (C) Bivariate tiled histogram of 3D vector directions. (D) Polar plots showing the frequency distribution of θ and ϕ angles throughout the volume 52

3.7 Calculated fluid-inspired metrics over 21 days in the presence of TGF-β1 of ring tissue development: (A) Streamlines, (B) (E_ρ) (light green) and (U_ρ) (dark green), and (C) Tortuosity values highlighting architectural changes 53

List of Tables

2.1 Values of U and E, along with the γ-distribution parameters γ_a and γ_b derived from τ histograms, are obtained for phantom images, bovine tendon, cervical porcine bone, and rat vaginal tissue. 31

3.1 Summary of E_ρ and U_ρ calculations for phantom and ring-shaped tissues 54

Preface

This dissertation is the culmination of several years of research, learning, and personal growth. It reflects not only the scientific questions I set out to answer but also the journey I undertook in becoming a researcher. The work presented here explores the spatial organization of collagen type I fibers and introduces computational tools, particularly the Fluid-Inspired Fiber Analysis (FIFA) framework that aim to make fiber analysis more interpretable and accessible.

My motivation for this work stemmed from a deep fascination with the extracellular matrix and its critical role in development, disease, and tissue mechanics. As I became immersed in the complexity of collagen morphology, particularly in soft tissues such as vaginal tissue and its biological interpretation, I developed a strong appreciation for the need to bridge experimental imaging, computational modeling, and biomechanical insight.

Throughout this process, I have been fortunate to work with brilliant mentors, collaborators, and peers who have shaped this project and my growth in countless ways. This thesis is both a product of my own efforts and a reflection of the community of support around me.

The research was conducted in the School of Engineering at Brown University, under the guidance of Kimani C. Toussaint Jr. Portions of this work have been presented at scientific conferences and prepared for publication.

The chapters are organized as follows: Chapter 1 provides background information;

Chapter 2 presents our work using fluid-inspired fiber analysis to study 2D images, Chapter 3 details the expansion of this approach to 3D datasets and Chapter 4 outlines directions for future work.

It is my hope that the tools and ideas developed in this thesis will continue to evolve and contribute to the broader understanding of matrix biology and fiber organization.

CHAPTER 1

Background

1.1 Extracellular Matrix

While cells are central to tissue function, they rely on a surrounding non-cellular framework called the extracellular matrix (ECM) to shape their environment and coordinate physiological processes essential for tissue development, maintenance and repair [1]. The ECM regulates cell adhesion, migration, proliferation and differentiation by providing essential biochemical and mechanical signals through interactions with cell surface receptors. It similarly influences intracellular pathways that control gene expression, survival and apoptosis. Throughout development and wound healing, it undergoes dynamic remodeling guiding tissue morphogenesis and vessel formation, angiogenesis. Additionally, the ECM modulates inflammatory responses and facilitates immune cell recruitment.

While fundamentally, the ECM is composed of water, proteins and polysaccharides, each tissue develops its unique ECM through dynamic interchange between cellular components and the microenvironment. This ECM-directed morphological organization gives rise to tissue-specific mechanical and biochemical properties that vary based on tissues's function and physiological state. The ECM is a concoction of two main macromolecules: fibrous proteins and proteoglycans. The principal fibrous proteins are collagens, fibronectins,

laminins and elastins. They provide structure and mechanical support. Proteoglycans are composed of a core protein covalently linked to glycosaminoglycan (GAG) chains [2].They offer unique hydration, buffering, binding and force resistant properties.

There are two major ECM types, interstitial a.k.a. connective tissue ECM and pericellular matrices [3, 4]. The interstitial ECM surrounds cells and is found in connective tissues such as skin, tendons, cartilage, bone and muscle. It interconnects the cells, guarantees structural integrity and modulates cell differentiation as well as migration [5]. The pericellular ECM is in direct contact with cells, acting as a selective barrier and a platform for cellular communication. It provides structural support and plays key roles in regulating cell polarity, differentiation, and filtration. To fulfill these roles, the ECM must engage in constant, bidirectional communication with cells, allowing them to sense and respond to mechanical and biochemical cues. Cells embedded into ECMs actively interact with this macromolecular network through a variety of surface receptors, including discoidin domain receptors, integrins, hyaluronan receptor CD44 and cell surface proteoglycans [3]. These receptors mediate the bidirectional exchange of signals between the cell and its surrounding environment. ECM signaling guides cells function and behavior. Cells are not only influenced by the ECM but also contribute to its composition; virtually all cell types synthesize, secrete, and remodel ECM macromolecules, making the matrix a dynamic and cell-regulated system [6].

Changes in the composition and structure of the ECM affect its biomechanical properties, structure and cell signaling thereby modulating their responses. ECM remodeling occurs during physiological conditions and in the presence of pathologies. Remodeling involves not only changes in ECM composition but also the enzymatic degradation and modification of its components, including collagens, proteoglycans, and glycoproteins. Key enzymes involved in ECM turnover include matrix metalloproteinases (MMPs), the ADAMTS (a disintegrin and metalloproteinase with thrombospondin motifs) family, and serine proteases such as plasmin [6]. These enzymes are regulated at multiple levels: tran-

scriptionally through cell- and time-specific gene expression, and post-transcriptionally through spatial localization and activation at subcellular or extracellular sites. The dynamic remodeling of the ECM plays essential roles in processes such as organogenesis, skeletal development, and stem cell differentiation, and its dysregulation is a hallmark of various diseases, particularly fibrotic disorders and cancer.

1.2 Type I Collagen

Among the various components of the extracellular matrix, collagen type I stands out as the most abundant and structurally significant fibrillar collagen. Its pivotal role in determining tissue architecture, mechanical properties, and cellular signaling makes it a central focus for understanding ECM function in both health and disease. Type I Collagen (Col-I) is composed of three polypeptide chains called alpha chains which twist together to form a triple helix. Col-I alpha 1 and alpha 2 chain synthesis starts at the transcriptions of COL1A1 and COL1A2 genes respectively. After mRNA processing, the mature mRNA is translated in the cytoplasm by ribosomes bound to the rough endoplasmic reticulum (ER) which produces preprocollagen alpha chains. In ER, preprocollagen is converted to procollagen by cleavage of its signal peptide. A procollagen results from two alpha-1 chains and one alpha-2 chain assembled into a triple helix. Procollagen molecules are then transported to the Golgi apparatus where they undergo final glycosylation, quality control and packaging. Here, procollagen has a diameter of about 300 nm.

As they exit the cell, procollagen terminal amino acid sequences which prevent spontaneous assembly are enzymatically removed transforming them into collagen molecules. Assembly into collagen fibrils starts during exocytosis. Fibrils can be up to 1 cm in length and up to about 500 nm in diameter. Through transmission electron microscopy, it has been observed that fibrils feature a D-periodic with D=67 nm which shows the arrangement of tropocollagen molecules [7]. Since tropocollagen molecules are about 300 nm long, the observed D-period shows that tropocollagen molecules don't align perfectly.

Each molecule is arranged by an offset of 67 nm. Many researchers groups made efforts to describe the three-dimensional structure of Col-I. Using an electron density map of Col-I, the data confirm that microfibrils have a quasi-hexagonal unit cell that forms a super twisted spiral structure resulting in a mature collagen fibril [8]. Fibrils covalently crosslinked with the aid of lysyl oxidase, aggregate laterally and bundle longitudinally to form collagen fibers. A certain degree of crosslinking is favorable for the mechanical properties of Col-I but excessive crosslinking results in brittle structures, a common symptom of aging [9].

The resulting collagen fiber architecture features such as shape, size, alignment, waviness and organization influences the mechanical, pathological behavior of collagen fibers. The microstructure of collage determines stiffness, extensibility and viscoelasticity of the ECM. The elastic modulus of a single collagen fiber is between 0.1 to 0.36 GPa [10]. The mechanical properties of collagen structures can be improved by crosslinking bonds. The force required to break the bonds is about 400 pN whose addition results in a stronger fiber. The crosslinking type, degree of crosslinking and strength of the crosslink influence fiber properties such as strength and rigidity. The orientation of collagen fibers influences the mechanical properties of the fibers. As tissues are subjected to diverse mechanical stresses, it results in a diversity of collagen architectures that satisfies the physiological requirements of tissues. The arrangement and density of Col-I fibrils enable the formation of ECMs with exceptionally high stiffness, often exceeding hundreds of MPa. The orientation of collagen fibers influences the overall stiffness of the extracellular matrix and modulates the rigidity of the fibronectin network, thereby affecting mechanotransduction and stiffness sensing through fibronectin-binding integrins [11]. Tissues where Col-I fibers are wavy or crimped are flexible and elastic. In disease, this structure is lost and fibers are straight and extended which increases cellular contractility and mechanical load.

Cell migration plays a critical role in development, wound healing, immune responses, and pathological processes such as cancer. While migration can occur randomly, it is

often guided along the long axis of collagen type I (Col-I) fibers. As the organization of Col-I fibers increases, cells become more elongated and exhibit contact guidance through integrin-mediated adhesion to the aligned matrix [12]. Highly aligned Col-I fibers promote cell spreading and directional migration, whereas disorganized Col-I networks support spreading without imparting directional cues. Notably, the influence of fiber orientation on migration is also cell-type dependent. For instance, aligned Col-I fibers have been shown to regulate mesenchymal stem cell (MSC) differentiation [13] and facilitate epithelial-to-mesenchymal transition (EMT) [14]. In addition to alignment, fiber diameter significantly affects ECM density and, consequently, cellular behavior. Smaller Col-I fiber diameters in native ECM lead to denser networks, while larger diameters form more open matrices. These structural differences influence cell phenotype: larger diameters are associated with a mesenchymal phenotype which are characterized by increased spreading, migration, and invasiveness. While smaller diameters support more epithelial phenotypes which are associated with compact, and less migratory behavior [14].

Given the significant influence of collagen fiber architecture on cell behavior, it is important to understand the factors that govern the shape and organization of these fibers within the extracellular matrix. The organization of collagen fibers is shaped by cellular forces, biochemical cues, mechanical loading, environmental conditions, and collagen composition, all of which interact to produce tissue-specific extracellular matrix architectures. The mechanical environment impacts the fiber orientation. Tension in the environment promotes straightness of the fibers [15] and fiber alignment along the direction of the applied force [16]. Moreover, variations in mechanical loading strength and tensile speed can influence the morphology and distribution of collagen fibers [17]. In vivo, tension is generated through the binding of integrins to collagen fibers. Additionally, shear forces have been shown to proportionally increase the degree of fiber orientation [13]and influence the direction of the fibers [18]. The shear rate also influences the length of the fibers, low shear rates translate into high fiber lengths.

Biochemical cues, including crosslinking agents and ECM components, play a critical role in regulating collagen fiber organization. Growth factors such as transforming growth factor-beta (TGF-β1) and platelet-derived growth factor (PDGF) influence both collagen synthesis and the expression of crosslinking enzymes. TGF-β1 stimulates fibroblasts to produce type I collagen (Col-I), while PDGF enhances the synthesis of fibronectin, hyaluronic acid, and collagenase. Both TGF-β1 and PDGF when secreted by macrophages have been shown to regulate α1(I) collagen and lysyl oxidase expression in lung fibroblasts [19]. Lysyl oxidase-mediated crosslinking of collagen fibrils contributes to fiber stabilization and affects their mechanical properties. In contrast, matrix MMPs, particularly MMP-1, MMP-8, and MMP-13, degrade and remodel Col-I fibers, thereby modulating fiber spacing and morphology. Proteoglycans and GAGs also contribute to fiber architecture. Decorin, a small leucine-rich proteoglycan, binds to Col-I fibrils, limiting their lateral growth and regulating interfibrillar spacing. Hyaluronan, a non-sulfated GAG, facilitates tissue hydration and spatial separation of collagen fibers, supporting a more organized and dynamic ECM structure.

Col-I fiber organization plays a critical role in many pathological conditions such as fibrosis, cancer, osteogenesis imperfecta and tendinopathy. Fibrosis is the end stage of chronic inflammatory reaction as a result of autoimmune reaction, persistent infections, allergic responses, chemical exposure, radiation, and tissue injury. Fibrotic diseases include cardiovascular fibrosis, systemic sclerosis, idiopathic pulmonary fibrosis, liver cirrhosis and progressive kidney disease. The primary cellular driver of fibrosis is the myofibroblast that when activated is the main collagen synthesis cell. Here, collagen production exceeds the rate at which it is degraded resulting in an increase of total collagen over time. These cells can be obtained from mesenchymal, endothelial and epithelial cells. Fibrotic disorders are commonly driven by a persistent irritant that maintains the production of growth factors, proteolytic enzymes, angiogenic signals, and fibrogenic cytokines. These mediators promote the excessive deposition of connective tissue components, leading to progressive

remodeling and disruption of normal tissue architecture [20]. In early or mild fibrosis, collagen fibers are disorganised while in advanced fibrosis, fibers are aligned depending on the tissue tension and fibroblast orientation [21]. This change often results in loss of tissue function (e.g. heart beating, air exchange in the lungs).

During cancer progression, Col-I fibers undergo remodeling that influence tumor behavior, invasion, metastasis and stiffness of the tumor microenvironment. Carcinomas with benign prognosis present larger collagen fibers that limit tumor growth. In contrast, malignant carcinomas show short and fragmented fibers. This change in size guides the metastatic invasion of tumor cells serving as a highway that facilitates migration of cancer cells to other locations with great vascularization [22]. If the healthy environment of collagen fibers presents wavy, relaxed or randomly aligned fibers, it has been observed that fibers become straight and aligned in the presence of cancer cells. Additionally, there is an increase in fiber crosslinking which leads to a bigger fiber diameter, increases in microenvironment stiffness and promotes focal adhesion [23]. Osteogenesis imperfecta results from mutations in Col-I genes (COL1A1, COL1A2), which profoundly disrupt collagen molecular assembly, fibrillogenesis, and fiber organization. These mutations lead to irregular fiber diameters [24], poorly packed and misshapen type I collagen fibrils, and disorganized fibrillar arrays [25]. Additional structural abnormalities include impaired crosslinking, reduced crimping, and randomly oriented fibers. Similarly, tendinopathy encompasses a range of pathological changes in tendons that lead to pain and impaired function. Healthy tendon consists of densely packed, highly organized Col-I fibers with sparse cellularity. In contrast, tendinopathic tendons exhibit fragmented and disorganized collagen fibers, where collagen degradation exceeds synthesis. This is accompanied by increased glycosaminoglycan content, neovascularization, and neoinnervation, all of which contribute to altered mechanical properties [26]. Therefore, the organization of Col-I fibers may serve as a valuable biomarker for disease. More broadly, this reflects a fundamental principle: the structural organization of both extracellular and intracellular

components is closely linked to tissue function and physiological health. For instance, the organization of sarcomeres in stem cell-derived cardiomyocytes (hiPSC-CMs) models is a critical marker of cardiomyocyte maturity and contractile function. Sarcomeres, the repeating contractile units of myofibrils, need to be properly aligned and periodic to enable effective force generation and synchronized electrical propagation. Over 85% of human pluripotent stem cell-derived cardiomyocytes (hiPSC-CMs) models report morphology and sarcomere organization quantification [27]. HiPSC-CMs often exhibit immature or disorganized sarcomeric patterns, limiting their physiological relevance. Consequently, assessing sarcomere architecture has become essential for evaluating cardiac differentiation protocols, disease modeling accuracy, and the functional readiness of engineered cardiac tissues.

1.3 Imaging

Understanding microstructural organization is essential for explaining physiological processes and maintaining cell viability. Local geometric cues play a critical role in regulating cell growth and survival during tissue development and in preserving microenvironmental balance. To visualize and quantify the organization of extracellular collagen networks, researchers commonly employ advanced imaging techniques. Col-I exhibits multiscale, hierarchical organization, with structural changes occurring at the molecular (collagen monomer), fibrillar, fibrous, and tissue levels. A variety of imaging techniques have been employed to investigate these structural changes across scales. At the nanometer scale, techniques such as transmission electron microscopy (TEM), scanning electron microscopy (SEM), cryo-electron microscopy (Cryo-EM) and atomic force microscopy (AFM) are used to visualize individual fibrils and fine ultrastructural details. At the subnanometer scale, micro-computed tomography (MicroCT) and X-ray diffraction. At the micrometer scale, where collagen fibers can be resolved, methods including second harmonic generation (SHG) microscopy, confocal microscopy, and polarized light microscopy (PLM)

are commonly used. Finally, at the tissue scale (tens of micrometers to millimeters), techniques such as optical coherence tomography (OCT) enable visualization of larger collagen architectures and spatial distribution across whole tissue sections.

Imaging the ultrastructure of collagen using TEM enables visualization of the ∼67 nm D-banding pattern, fibril diameter, and packing. TEM offers high resolution (∼0.1-1 nm) but requires ultrathin sections (∼50-100 nm) and extensive sample preparation, including fixation and dehydration, which can alter the native collagen structure. Imaging occurs in a vacuum, making it incompatible with hydrated or live samples. SEM is commonly used to examine the surface morphology and fibril orientation of collagen type I. It provides resolution around 1-10 nm but also requires sample dehydration and coating with conductive materials such as gold. While useful for topography, SEM lacks the resolution to resolve sub-fibrillar features as clearly as TEM. Neither TEM nor SEM can image collagen in its fully hydrated state. Cryo-EM has emerged as a powerful electron microscopy technique where sample hydration is maintained near physiological state. Its in situ resolution is (∼2-20 nm) and it can resolve near atomic resolution 2-4 Angstroms for homogeneous samples using single particle analysis [28]. For Col-I, Cryo-EM allows researchers to study interaction with other EM components and its hierarchical assembly at its native state. Similarly, AFM allows imaging of individual fibrils, D-banding, and mechanical properties such as Young's modulus, stiffness, and adhesion. AFM can operate in air or liquid, preserving native hydration. It offers resolution of ∼1-5 nm but is limited to surface imaging over a small scan area. Additionally, AFM is slower and more sensitive to vibration, and samples must be flat with well-adhered fibrils.

At the um scale SHG microscopy is considered the gold standard for imaging Col-I fiber structure without the need of labels. SHG microscopy, a second-order nonlinear optical technique, is widely used to visualize and quantify collagen type I (Col-I) fiber morphology in tissues with lateral resolution of (200-500 nm) and axial resolution between (0.5-1.5 um) [29, 30]. Col-I generates a strong and highly specific SHG signal due to its permanent

dipole moment and high hyperpolarizability, enabling robust, label-free imaging. Collagen natural birefringence allows for label-free imaging using PLM. Staining with picrosirius red enhances the birefringence and contrast. This technique is widely used because of its simplicity and low cost. Its resolution is 1-2 um and often 2D alone. Samples are required to be (5-10 um) thick for proper light transmission. Confocal microscopy has similar resolutions as SHG. It provides optical sectioning but it is dependent on fluorescence labeling (Fast green, picrosirius red). At the macroscale level, OCT offers a non-invasive, real time imaging technique and deeper penetration. Its resolution is around (4-15 um) laterally and (10-20 um) axially. Penetration depth is 1-2 mm in the tissue. While contrast is not specific to collagen, it can be combined with polarization.

1.4 Fiber Quantification

While these imaging techniques reveal detailed structural features of collagen, quantitative analysis is essential to quantify fiber organization. A variety of quantitative SHG imaging techniques have been developed to investigate the relationship between collagen microstructure and biological function. These methods aim to quantify fiber density, fiber orientation, fiber anisotropy, fiber diameter and fiber length. The techniques quantify fiber orientation in 2D from microscopy images and in 3D from image z-stacks. Fiber orientation is then used to quantify fiber anisotropy. In 2D, fiber orientation computation includes transform-based methods including curvelet, wavelet, and Fourier transforms have been applied to quantify Col-I fiber orientation distribution [31, 32]. Other methods include Hessian matrix based methods and machine learning. In particular, Fourier transform-second harmonic generation (FT-SHG) analysis transforms an image from the spatial domain (e.g. pixels, edges, textures) into the frequency domain (harmonic functions) which describes repeating direction of repeating patterns. Any image can be reconstructed by a mix of sinusoidal waves of different frequencies. Here, the amplitude represents how prominent the wave is and the phase, the position of the wave. This analysis has been

used to determine fiber orientation of Col-I fibers from SHG images. A highly directional Fourier spectrum indicates aligned fibers and a more uniform spectrum on both axes shows randomly oriented fibers. In this technique, dark thresholds and isotropic thresholds are important to filter noise and competing orientations. The dominant frequency is perpendicular to the orientation of the original image. While calculating, the spread around the fitting line shows how uniform or varied fiber orientations are. The key is to divide the image in grids that capture the changes in fiber orientation. Results from this analysis are often represented as a frequency distribution where preferred orientation is reported.

Curvelet transform analysis is commonly known by two packages: CurveAlign and CT-FIRE. This analysis measures global and local fiber alignment. These techniques do not extract individual fibers, instead they use curvelets to obtain the edges of fibers where the information about their scale, location and orientation is present. It also measures angles with respect to a boundary of interest. It also calculates nearest neighbor alignment. Angle calculations are limited between 0 to 180 when calculated with respect to the horizontal and 0 to 90 for angles relative to a boundary. The alignment coefficient is between 0 to 1 and it indicates the dispersity of the fiber orientations. This term is also known across the field as anisotropy index, coherence, anisotropy, coherence, circular variance and mean vector length. CurveAlign limitations include difficulty to extract intact fibers from wavy shapes that have varying intensities in dense fiber environments resulting in fiber segments. This algorithm relies on even fiber thicknesses. Computation times for large images could be long. CT-FIRE could be limited to fibers with highest intensities and might be sensitive to fiber thickness changes. This tool is limited to 2D analysis [33].

Hessian matrix based methods use the second-order derivatives to identify line-like structures from fibers and their orientations [34]. Here, lines can be detected with subpixel accuracy. This algorithm detects thin or thick fibers using scale-space analysis,

it uses second derivatives to find line centers, it facilitates threshold finding by inputting semantically meaningful values and features are found based on intensity structure in the image. Col-I segmentation has been studied with the use of deep learning convolutional neural networks. This study improves fiber segmentation for SHG images with variable intensities [35]. It uses 138 836 image patches from which 70% are for training, 20% for validation and 10% for testing. Each patch is about 64 by 64 pixels and they have various depths between 130 to 170 um. Training takes about 92 min. This technique training is limited to the specific magnification of the input images. Deep learning-based collagen fiber centerline extraction is also trained with synthetic data that is generated with a variational autoencoder to increase generalizability [36]. This approach derives fiber orientation, alignment, density and length.

While a variety of techniques have been developed to quantify fiber organization in 2D, extending these analyses into 3D presents significant challenges. Fibers with steep elevation angles relative to the imaging plane often appear with nearly circular cross-sections, making their orientation difficult to resolve using 2D analysis alone [37]. Furthermore, many existing algorithms are restricted to either in-plane or out-of-plane orientation measurements, which provides only a partial view of fiber dispersion [38]. Even when both in-plane and out-of-plane information is available, these distributions cannot be directly combined to reconstruct a complete 3D orientation distribution, since the covariance between the two directions is not captured [39, 40] These limitations highlight the need for robust, fully 3D approaches that can capture both local variability and global organization of collagen architectures. A range of computational and imaging-based techniques has been developed to quantify the three-dimensional organization of fibrous networks such as collagen. Fourier transform-based approaches [37, 38, 41] analyze volumetric image data in frequency space to estimate orientation distribution functions, providing robust global measures of fiber alignment and anisotropy. Fiber orientation information is further analyzed using statistical techniques like spherical

variance and probability density function fitting parameters. Segmentation methods quantify collagen fiber orientation, branching and diameter [39, 40]. This method relies on preanalysis to determine the voxel of interest. Then, it calculates fiber orientation by fitting a line through the center of the voxel. This technique is sensitive to fiber diameter, density and image quality. The computational cost is higher for thinner fibers and dense fiber structures. Fiber orientation is further quantified with spherical histograms and a modified spherical variance that accounts for the nondirectional nature of fiber angles. Open-source frameworks such as OrientationPy use structure tensors to calculate fiber direction, energy and coherency at every pixel [42]. To further assess orientation, circular standard deviation for global as well as local direction and von Mises distribution are used. They also calculate fiber straightness by manually tracing the fibers. In addition to global orientation measures, post quantification of fiber orientation has been optimized for biological tissues. Automated quantification methods [43, 44] integrate voxel-wise orientation algorithms with modified directional variance feature extraction to evaluate fiber density, anisotropy, and spatial distribution, supporting analysis of collagen fibers in native as well as engineered tissues, fibroblast alignment and neural axons during traumatic brain injury. Similarly filament tracing algorithms such as SOAX [40] extract fiber centerlines and identify junctions in 3D fiber structures. It captures 3D orientation using a bivariate histogram and it captures curvature. This method is best for spreadout fibers where the spacing between neighboring filaments is larger than the microscope's point spread function.

While these techniques have advanced the field substantially, each has inherent limitations. Fourier-based approaches provide global orientation statistics but cannot resolve local microstructural variations. Segmentation and filament tracing methods allow detailed reconstructions but are computationally intensive and often sensitive to image quality, noise, or parameter tuning. Moreover, many approaches are limited to either 2D projections or rely on assumptions that reduce the accuracy of 3D orientation measurements in

thick, scattering samples. Most post analysis of collagen fiber orientations use statistical methods while missing spatial information.

These challenges highlight the need for methods that combine the robustness of global statistical analysis with the spatial precision of local fiber tracking, while remaining computationally efficient and adaptable to diverse tissue types. Addressing this gap motivated the development of the Fluid-Inspired Fiber Analysis (FIFA), a vector-field–based framework for quantifying collagen fiber organization in three dimensions. By drawing on fluid mechanics principles, FIFA provides a scalable approach that captures both local and global fiber architecture, enabling robust quantification in scattering biological tissues.

This thesis presented a framework for quantifying two and three-dimensional collagen fiber organization using Fourier Transform Second Harmonic Generation (FT-SHG) imaging in conjunction with the Fluid-Inspired Fiber Analysis (FIFA) for quantifying spatial changes in fiber architecture in 2D and 3D[45–47]. Our analysis demonstrates that collagen microarchitecture is not only structurally diverse but also encodes meaningful information about tissue state. The development of FIFA provides a new lens through which to assess fiber alignment, directionality, and deformation through spatially resolved metrics. These metrics energy(E), enstrophy (U) and tortuosity (τ) offer quantitative metrics of spatial changes in collage fiber organization and insights on waviness of the fibers.

References

[1] C. Frantz, K. M. Stewart, and V. M. Weaver, "The extracellular matrix at a glance," *J. Cell Sci.*, vol. 123, pp. 4195–4200, Dec. 2010.

[2] J. R. Couchman and C. A. Pataki, "An introduction to proteoglycans and their localization," *J. Histochem. Cytochem.*, vol. 60, pp. 885–897, Dec. 2012.

[3] A. D. Theocharis, S. S. Skandalis, C. Gialeli, and N. K. Karamanos, "Extracellular matrix structure," *Adv. Drug Deliv. Rev.*, vol. 97, pp. 4–27, Feb. 2016.

[4] A. D. Theocharis, D. Manou, and N. K. Karamanos, "The extracellular matrix as a multitasking player in disease," *FEBS J.*, vol. 286, pp. 2830–2869, Aug. 2019.

[5] J. Winkler, A. Abisoye-Ogunniyan, K. J. Metcalf, and Z. Werb, "Concepts of extracellular matrix remodelling in tumour progression and metastasis," *Nat. Commun.*, vol. 11, p. 5120, Oct. 2020.

[6] P. Lu, K. Takai, V. M. Weaver, and Z. Werb, "Extracellular matrix degradation and remodeling in development and disease," *Cold Spring Harb. Perspect. Biol.*, vol. 3, pp. a005058–a005058, Dec. 2011.

[7] M. D. Shoulders and R. T. Raines, "Collagen structure and stability," *Annu. Rev. Biochem.*, vol. 78, pp. 929–958, July 2009.

[8] J. P. R. O. Orgel, T. C. Irving, A. Miller, and T. J. Wess, "Microfibrillar structure of type I collagen in situ," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, pp. 9001–9005, June 2006.

[9] M. J. Buehler, "Nature designs tough collagen: explaining the nanostructure of collagen fibrils," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 103, pp. 12285–12290, Aug. 2006.

[10] P. Dutov, O. Antipova, S. Varma, J. P. R. O. Orgel, and J. D. Schieber, "Measurement

of elastic modulus of collagen type I single fiber," *PLoS One*, vol. 11, p. e0145711, Jan. 2016.

[11] V. W. Tang, "Collagen, stiffness, and adhesion: the evolutionary basis of vertebrate mechanobiology," *Mol. Biol. Cell*, vol. 31, pp. 1823–1834, Aug. 2020.

[12] A. D. Doyle, N. Carvajal, A. Jin, K. Matsumoto, and K. M. Yamada, "Local 3D matrix microenvironment regulates cell migration through spatiotemporal dynamics of contractility-dependent adhesions," *Nat. Commun.*, vol. 6, p. 8720, Nov. 2015.

[13] B. Lanfer, U. Freudenberg, R. Zimmermann, D. Stamov, V. Körber, and C. Werner, "Aligned fibrillar collagen matrices obtained by shear flow deposition," *Biomaterials*, vol. 29, pp. 3888–3895, Oct. 2008.

[14] A. Ravikrishnan, T. Ozdemir, M. Bah, K. A. Baskerville, S. I. Shah, A. K. Rajasekaran, and X. Jia, "Regulation of epithelial-to-mesenchymal transition using biomimetic fibrous scaffolds," *ACS Appl. Mater. Interfaces*, vol. 8, pp. 17915–17926, July 2016.

[15] J. Lin, Y. Shi, Y. Men, X. Wang, J. Ye, and C. Zhang, "Mechanical roles in formation of oriented collagen fibers," *Tissue Eng. Part B Rev.*, vol. 26, pp. 116–128, Apr. 2020.

[16] D. Vader, A. Kabla, D. Weitz, and L. Mahadevan, "Strain-induced alignment in collagen gels," *PLoS One*, vol. 4, p. e5902, June 2009.

[17] L. Han, J. Lin, C. Du, C. Zhang, X. Wang, and Q. Feng, "Effect of mechanical microenvironment on collagen self-assembly in vitro," *J. Funct. Biomater.*, vol. 14, p. 235, Apr. 2023.

[18] N. Saeidi, E. A. Sander, and J. W. Ruberti, "Dynamic shear-influenced collagen self-assembly," *Biomaterials*, vol. 30, pp. 6581–6592, Dec. 2009.

[19] R. Koslowski, D. Seidel, E. Kuhlisch, and K.-P. Knoch, "Evidence for the involvement of TGF- and PDGF in the regulation of prolyl 4-hydroxylase and lysyloxidase in cultured rat lung fibroblasts," *Exp. Toxicol. Pathol.*, vol. 55, pp. 257–264, Jan. 2003.

[20] T. A. Wynn, "Cellular and molecular mechanisms of fibrosis," *J. Pathol.*, vol. 214, pp. 199–210, Jan. 2008.

[21] A. Hervas, A. Ruiz-Sauri, E. de Dios, M. J. Forteza, G. Minana, J. Nunez, C. Gomez, C. Bonanad, N. Perez-Sole, J. Gavara, F. J. Chorro, and V. Bodi, "Inhomogeneity of collagen organization within the fibrotic scar after myocardial infarction: results in a swine model and in human samples," *J. Anat.*, vol. 228, pp. 47–58, Jan. 2016.

[22] A. P. Vargas Garcia, L. A. Reis, B. R. M. Ribeiro, C. B. Nunes, A. M. de Paula, and G. D. Cassali, "Comparative evaluation of collagen modifications in breast cancer in human and canine carcinomas," *Sci. Rep.*, vol. 14, p. 28846, Nov. 2024.

[23] K. R. Levental, H. Yu, L. Kass, J. N. Lakins, M. Egeblad, J. T. Erler, S. F. T. Fong, K. Csiszar, A. Giaccia, W. Weninger, M. Yamauchi, D. L. Gasser, and V. M. Weaver, "Matrix crosslinking forces tumor progression by enhancing integrin signaling," *Cell*, vol. 139, pp. 891–906, Nov. 2009.

[24] H. Stöss and P. Freisinger, "Collagen fibrils of osteoid in osteogenesis imperfecta: morphometrical analysis of the fibril diameter," *Am. J. Med. Genet.*, vol. 45, p. 257, Jan. 1993.

[25] N. Bishop, "Bone material properties in osteogenesis imperfecta: Bone material properties in oi," *J. Bone Miner. Res.*, vol. 31, pp. 699–708, Apr. 2016.

[26] N. L. Millar, K. G. Silbernagel, K. Thorborg, P. D. Kirwan, L. M. Galatz, G. D. Abrams, G. A. C. Murrell, I. B. McInnes, and S. A. Rodeo, "Tendinopathy," *Nat. Rev. Dis. Primers*, vol. 7, p. 1, Jan. 2021.

[27] J. K. Ewoldt, S. J. DePalma, M. E. Jewett, M. Karakan, Y.-M. Lin, P. Mir Hashemian, X. Gao, L. Lou, M. A. McLellan, J. Tabares, M. Ma, A. C. Salazar Coariti, J. He, K. C. Toussaint, T. G. Bifano, S. Ramaswamy, A. E. White, A. Agarwal, E. Lejeune, B. M. Baker, and C. S. Chen, "Induced pluripotent stem cell-derived cardiomyocyte

in vitro models: benchmarking progress and ongoing challenges," *Nat. Methods*, pp. 1–17, Nov. 2024.

[28] T. Nakane, A. Kotecha, A. Sente, G. McMullan, S. Masiulis, P. M. G. E. Brown, I. T. Grigoras, L. Malinauskaite, T. Malinauskas, J. Miehling, T. Uchański, L. Yu, D. Karia, E. V. Pechnikova, E. de Jong, J. Keizer, M. Bischoff, J. McCormack, P. Tiemeijer, S. W. Hardwick, D. Y. Chirgadze, G. Murshudov, A. R. Aricescu, and S. H. W. Scheres, "Single-particle cryo-EM at atomic resolution," *Nature*, vol. 587, pp. 152–156, Nov. 2020.

[29] X. Chen, O. Nadiarynkh, S. Plotnikov, and P. J. Campagnola, "Second harmonic generation microscopy for quantitative analysis of collagen fibrillar structure," *Nat. Protoc.*, vol. 7, pp. 654–669, Mar. 2012.

[30] M. Kabir, H. Rajput, V. Kelkar, A. Salazar Coariti, and K. Toussaint, "Demonstration of flat-top beam illumination in widefield multiphoton microscopy," *J. Biomed. Opt.*, vol. 25, pp. 1–8, Nov. 2019.

[31] R. A. R. Rao, M. R. Mehta, and K. C. Toussaint, Jr, "Fourier transform-second-harmonic generation imaging of biological tissues," *Opt. Express*, vol. 17, pp. 14534–14542, Aug. 2009.

[32] J. A. McGuire, J. L. Monclova, A. C. S. Coariti, C. A. Stine, K. C. Toussaint, Jr, J. M. Munson, D. A. Dillard, and R. De Vita, "Tear propagation in vaginal tissue under inflation," *Acta Biomater.*, vol. 127, pp. 193–204, June 2021.

[33] Y. Liu, A. Keikhosravi, C. A. Pehlke, J. S. Bredfeldt, M. Dutson, H. Liu, G. S. Mehta, R. Claus, A. J. Patel, M. W. Conklin, D. R. Inman, P. P. Provenzano, E. Sifakis, J. M. Patel, and K. W. Eliceiri, "Fibrillar collagen quantification with curvelet transform based computational methods," *Front. Bioeng. Biotechnol.*, vol. 8, p. 198, Apr. 2020.

[34] C. Steger, "An unbiased detector of curvilinear structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 2, pp. 113–125, 1998.

[35] A. E. Woessner and K. P. Quinn, "Improved segmentation of collagen second harmonic generation images with a deep learning convolutional neural network," *J. Biophotonics*, p. e202200191, Sept. 2022.

[36] H. Park, B. Li, Y. Liu, M. S. Nelson, H. M. Wilson, E. Sifakis, and K. W. Eliceiri, "Collagen fiber centerline tracking in fibrotic tissue via deep neural networks with variational autoencoder-based synthetic training data generation," *Med. Image Anal.*, vol. 90, p. 102961, Dec. 2023.

[37] R. Alberini, A. Spagnoli, M. J. Sadeghinia, B. Skallerud, M. Terzano, and G. A. Holzapfel, "Fourier transform-based method for quantifying the three-dimensional orientation distribution of fibrous units," *Sci. Rep.*, vol. 14, p. 1999, Jan. 2024.

[38] Y. Liu, A. Keikhosravi, G. S. Mehta, C. R. Drifka, and K. W. Eliceiri, "Methods for quantifying fibrillar collagen alignment," *Methods Mol. Biol.*, vol. 1627, pp. 429–451, 2017.

[39] J. D. Eekhoff and S. P. Lake, "Three-dimensional computation of fibre orientation, diameter and branching in segmented image stacks of fibrous networks," *J. R. Soc. Interface*, vol. 17, p. 20200371, Aug. 2020.

[40] T. Xu, D. Vavylonis, F.-C. Tsai, G. H. Koenderink, W. Nie, E. Yusuf, I-Ju Lee, J.-Q. Wu, and X. Huang, "SOAX: a software for quantification of 3D biopolymer networks," *Sci. Rep.*, vol. 5, p. 9081, Mar. 2015.

[41] W. Lee, A. O. Moghaddam, Z. Lin, B. L. McFarlin, A. J. Wagoner Johnson, and K. C. Toussaint, "Quantitative classification of 3D collagen fiber organization from volumetric images," *IEEE Trans. Med. Imaging*, vol. 39, pp. 4425–4435, Dec. 2020.

[42] R. Rezakhaniha, A. Agianniotis, J. T. C. Schrauwen, A. Griffa, D. Sage, C. V. C. Bouten, F. N. van de Vosse, M. Unser, and N. Stergiopulos, "Experimental investigation of collagen waviness and orientation in the arterial adventitia using confocal

laser scanning microscopy," *Biomech. Model. Mechanobiol.*, vol. 11, pp. 461–473, Mar. 2012.

[43] Z. Liu, D. Pouli, D. Sood, A. Sundarakrishnan, C. K. Hui Mingalone, L. M. Arendt, C. Alonzo, K. P. Quinn, C. Kuperwasser, L. Zeng, T. Schnelldorfer, D. L. Kaplan, and I. Georgakoudi, "Automated quantification of three-dimensional organization of fiber-like structures in biological tissues," *Biomaterials*, vol. 116, pp. 34–47, Feb. 2017.

[44] Z. Liu, K. P. Quinn, L. Speroni, L. Arendt, C. Kuperwasser, C. Sonnenschein, A. M. Soto, and I. Georgakoudi, "Rapid three-dimensional quantification of voxel-wise collagen fiber orientation," *Biomed. Opt. Express*, vol. 6, pp. 2294–2310, July 2015.

[45] A. C. Salazar Coariti, M. Fabien, J. Guzmán, and K. Toussaint, "Quantitative second-harmonic generation imaging analysis based on fluid-dynamics measures," *Frontiers in Optics*, Sept. 2020.

[46] A. C. Salazar Coariti, M. S. Fabien, J. Guzman, J. A. McGuire, R. De Vita, and K. C. Toussaint, "Fluid mechanics approach to analyzing collagen fiber organization," *J. Biomed. Opt.*, vol. 27, Jan. 2022.

[47] A. S. Coariti, N. Gaitainis, C. Hopkins, J. R. Morgan, and K. Toussaint, "Fluid-inspired fiber analysis of 3D collagen architecture," *Available at SSRN 5255827*.

CHAPTER 2

Fluid Mechanics Approach to Analyzing 2D Collagen Fiber Organization

2.1 Introduction

The alignment of collagen fibers is essential to maintain tissue mechanical integrity, direct cell behavior, and is a biomarker of healthy function and pathological remodeling [1–4]. While the previous chapter focused on methods for obtaining fiber orientation, orientation analysis alone does not fully capture the degree of fiber alignment. Therefore, statistical and graphical methods are commonly employed to quantify orientation distribution, anisotropy, and preferred directionality in fibrillar collagen-rich tissues. Fiber orientation is usually visualized by a histogram of orientations or a rose plot [5, 6]. The histogram of orientations shows the frequency distribution of fiber angles, and polar plot offers a circular visualization of fiber angles.

Several metrics are commonly used to quantify the alignment of collagen fibers, including circular statistical parameters such as mean angle, mean vector length, circular standard

deviation and circular variance [3, 7, 8]. Fiber orientation is inherently periodic rather than linear[9]. Once it is recognized that 360° is actually nearer to 5° than to 330°, it becomes evident that circular data require a distinct statistical approach compared to linear variables. Many circular statistics methods assume data are centered around a dominant direction, unimodal. However, complex biological tissues have multiple dominant directions. Additionally, most circular statistics used in collagen fiber analysis are axial in nature, meaning they treat orientations such that a fiber pointing in one direction (e.g., 0°) is considered equivalent to one pointing in the opposite direction (e.g., 180°). Thus, they assume there is no inherent direction. In biological soft tissues, crimped fibers are commonly observed, and capturing their continuous structure requires orientation data with defined directionality. In addition, circular statistics typically treat each directional data point as independent of its spatial position. Spatial changes in collagen architecture are crucial because they reveal pathological remodeling and allow a sensitive assessment of structural integrity and disease progression.

Spatial fiber organization complexity can be quantified by reinterpreting the problem within the framework of a vector field (VF) and utilizing fluid mechanics quantification metrics. VF analysis has been employed to study a wide range of phenomena, including human mobility patterns, fiber reconstruction, and collective bacterial motion. For example, one study used a mesoscopic VF to capture and simulate recurrent human movement, providing insights into epidemic spread and informing urban planning[10]. In another case, vector fields were used to trace nerve fiber pathways in the brain's white matter, with streamlines representing their global trajectories[11]. Additionally, VF analysis has been applied to characterize the collective behavior of bacteria, quantifying their fluid-like dynamics using the Reynolds number, which compares inertial and viscous forces[12]. These diverse applications highlight the versatility of VF analysis and its potential to offer novel insights into the spatial organization of collagen fibers.

In this chapter, we introduce a method for analyzing SHG collagen images by treating

them as pseudo-velocity fields (pseudo-VFs) and applying evaluation metrics inspired by fluid mechanics. Our approach begins with generating a two-dimensional quadrilateral mesh, where each cell contains a pseudounit vector representing the dominant local orientation, determined through FT-SHG analysis. This resulting vector field is comparable to a fluid flow velocity map. We then apply fluid mechanics measures such as enstrophy (E), tortuosity (τ), and energy (U) to quantify these pseudo-VFs. Within this framework, the metrics E and U characterize the collective behavior of orientation changes—vortex-like patterns and abrupt directional shifts, respectively, while τ serves as a measure of streamline waviness.

The structure of this chapter is as follows: Section 2.2 details the methodology and sample preparation used to evaluate fluid mechanics-based metrics in bovine tendon, cortical porcine bone, and rat vaginal tissue. Section 2.3 presents and discusses the results. Finally, Section 2.4 outlines the conclusions and directions for future research.

2.2 Materials and methods

Bovine tendon was immersed in phosphate-buffered saline (PBS) and cryosectioned at 200μm intervals using a Thermo Scientific NX50 cryostat. Sections were mounted on glass slides for second harmonic generation (SHG) imaging. Porcine femoral bones were stripped of soft tissue, wrapped in PBS, and frozen at -20°C. Prior to imaging, bones were thawed in PBS for 12 hours, transversely sectioned into mid-diaphyseal hollow cylinders, and progressively polished. Rat vaginal tissue was harvested from virgin female Sprague-Dawley rats post-decapitation and frozen at -20°C. Samples were later thawed and dehydrated in PBS at room temperature, then longitudinally sectioned along the urethral axis. The dorsal surface was flattened and secured to a PBS-filled Petri dish using cyanoacrylate adhesive. All samples were study using a commercial multi-photon microscope with excitation near 780 nm. Additional experimental details are provided in previous studies (see Refs. [2, 3, 7]). For the porcine and rat tissues, a minimum of

three representative SHG images were selected for analysis. Bovine samples were sourced from a local abattoir. Porcine tissue protocols and imaging were based on prior work [Ref. [2]], although unpublished image data were used in this study. Harvesting of rat vaginal tissues was approved by the Virginia Tech Institutional Animal Care and Use Committee.

In characterizing collagen fiber structure, we considered the parameters τ, U, and E. Acquired SHG images were converted into pseudo-VFs of unit length, representing the preferred fiber orientation as determined by 2D Fourier transform analysis[13].Dark regions identified through FT-SHG image analysis were interpreted as areas around which fluid might flow. To compute τ, which captures the degree of bending or twisting in fibers relative to a straight configuration, VFs were first converted into streamlines. In fluid mechanics, streamlines are lines tangent to the velocity vectors of a flow field. In our case, the streamlines did not perfectly replicate the original fibers due to the presence of dark (missing fiber) and isotropic (disorganized fiber) regions in the images. To address this, we initialized a streamline from each vector within a VF. The parameter τ was then calculated by quantifying the path length P of the streamlines tangent to the VF, relative to the straight-line distance D between their endpoints, as described in Ref. [14]. For the calculation of E and U, the VFs were evaluated using a 2D forward finite difference approach. Definitions of τ, E, and U are provided below.

$$\tau = \frac{P}{D}, \tag{2.1}$$

$$E = \int_A |\nabla \times \vec{v}|^2 \, dA, \tag{2.2}$$

and

$$U = \rho \int_A \nabla \cdot \vec{v} \, dA. \tag{2.3}$$

Here, $\vec{v}$ denoted the pseudo-VF field, ρ represented the fluid density that was set to

one, ∇ denotes the del operator, and A denoted the area encompassing the VFs. Since our data consisted of static images that do not capture actual velocity, we used a pseudo-VF to represent $\vec{v}$, reflecting changes in fiber orientation (i.e. direction) rather than true motion. We quantified the histogram of τ using the gamma distribution (γ), a frequency distribution defined by two parameters. The skewness of the distribution was inversely related to the shape parameter γ_a, while the spread was governed by the scale parameter γ_b. Fluid metrics E,U, and τ were calculated in MATLAB. An overview of the algorithm is provided in Fig. 1.

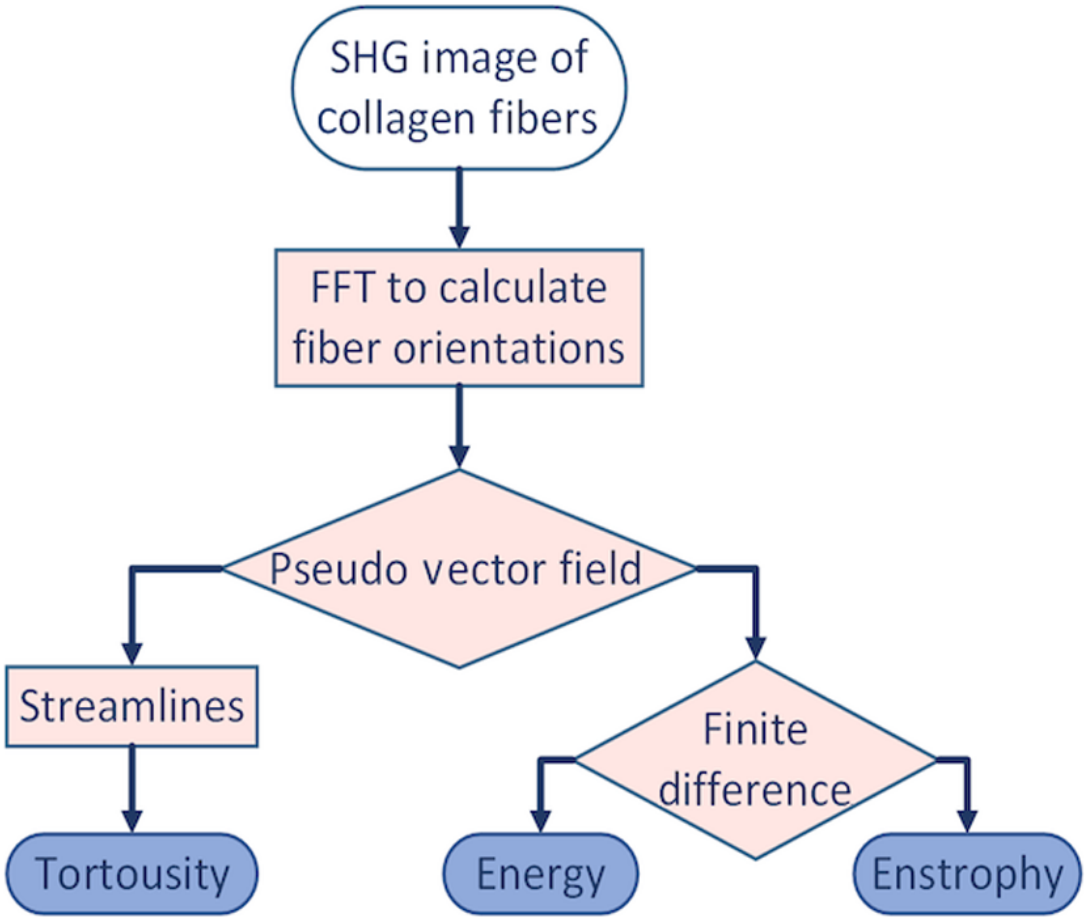

Figure 2.1: Diagram outlining the algorithm for calculating τ, U, and E

To calibrate our method, we generated digital phantoms in MATLAB. Figure 2(a, i) displays perfectly straight fibers, with the corresponding vector field distribution and streamline outputs shown in Figs. 2(a, ii) and 2(a, iii), respectively. As expected for

this idealized case, τ was found to be 1, as indicated by the histogram in Fig. 2(a, iv). Although this straight-fiber model served as a reference, it does not fully capture the behavior of natural collagen fibers, which, even when uniformly organized, typically lack rigid rod-like features. In Fig. 2(b, i), we present a simulated phantom of wavy fibers, with the associated VF and streamline distributions shown in Figs. 2(b, ii) and 2(b, iii). In this case, τ measured 1.25. For these phantoms, U and E were also calculated, yielding values of 0 for straight fibers and 191.9 and 106.0, respectively, for wavy fibers.

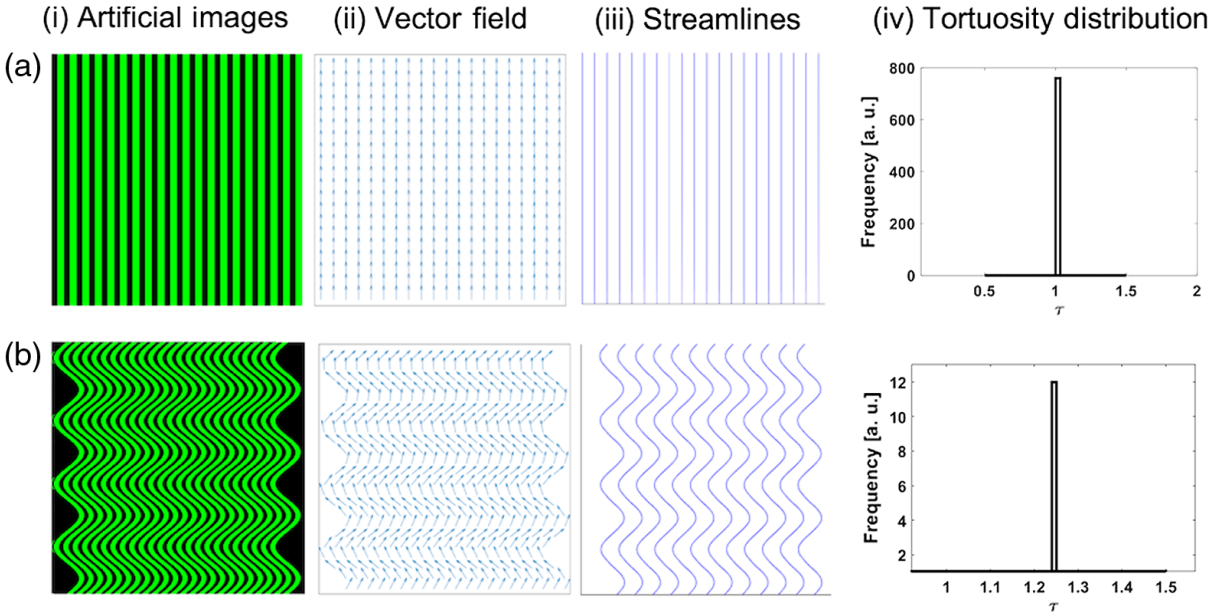

Figure 2.2: Representative synthetic images showing (a)(i) uniformly aligned fibers and (b)(i) wavy fibers, along with (ii) their corresponding vector field (VF) maps, (iii) the resulting streamlines, and (iv) histograms of the computed tortuosity values.

2.3 Results and Discussions

This study examines SHG images acquired from bovine tendon, rat vaginal tissue, and porcine cortical bone. Due to its relatively organized collagen architecture, bovine tendon serves as a representative tissue for assessing the algorithm's performance on structures composed of predominantly straight or in-plane wavy fibers. Fig.2.3(a) shows the application of the analysis to a bovine tendon SHG image, which contains a combination of organized and slightly undulated fibers. In Fig. 2.3(b) we show the VF-derived

streamlines and Fig. 2.3(c) presents the τ distribution ranging from 1.0 to 1.3, with values predominantly clustering around 1.1. As discussed previously, τ values near 1 are characteristic of straight fibers, while the presence of waviness increases the observed τ range. For this tissue, the computed values of U and E were 284 and 302, respectively. Although U and E share similar mathematical formulations, they offer distinct interpretive value: U captures abrupt changes in fiber orientation associated with crimped or wavy structures, whereas E is more sensitive to circular or vortex-like fiber organization.

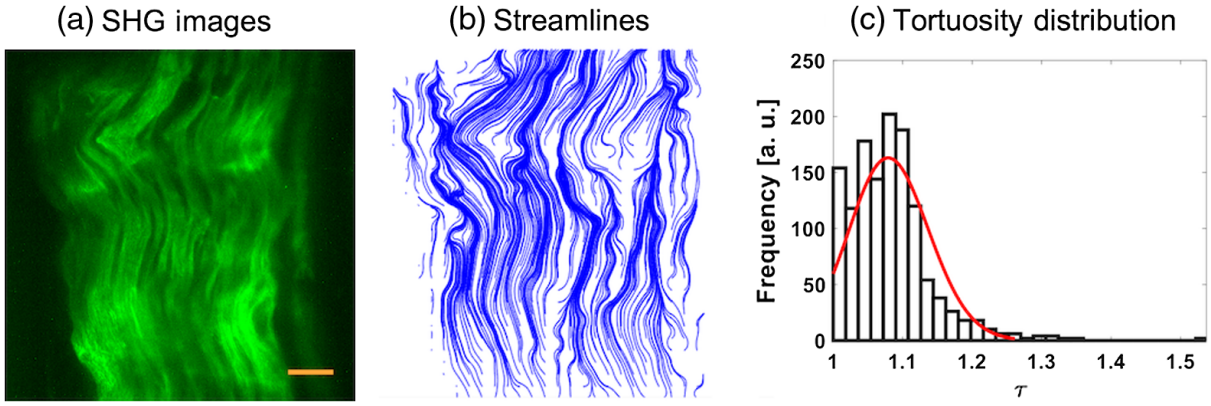

Figure 2.3: (a) Representative SHG image of bovine tendon, (b) corresponding streamline map derived from the vector field, and (c) distribution of tortuosity (τ) values. Scale bars correspond to 25µm.

We further apply the analysis to multiple SHG images of porcine cortical bone, including regions with uniformly aligned fibers Fig. 2.4(a, i) and regions exhibiting vortex-like organization associated with the presence of Haversian canals Fig. 2.4(b, i). The corresponding VF streamlines and τ distributions are shown in columns (ii) and (iii), respectively. As illustrated in Fig. 2.4(a, ii), the streamlines reflect the predominantly linear organization of collagen fibers in cortical bone, which is corroborated by the narrow τ distribution observed in Fig. 2.4(a, iii), ranging from 1.0 to approximately 1.007 and strongly centered around 1. These findings are consistent with the presence of primarily straight fibers. The calculated values for U and E in this region are $5.70 \times 10 - 1 \pm 0.097$ and $6.20 \times 101 \pm 0.067$, respectively. In contrast, images containing Haversian canals exhibit a wider τ distribution, ranging from 1.0 to 2.0, indicative of more complex fiber

organization. In these regions, the calculated U and E values are $2.58 \times 10 + 3 \pm 262.7$ and $2.64 \times 10 + 3 \pm 270.8$, respectively. The pronounced increase in both metrics reflects the circular, vortex-like fiber arrangements induced by the Haversian canal structures.

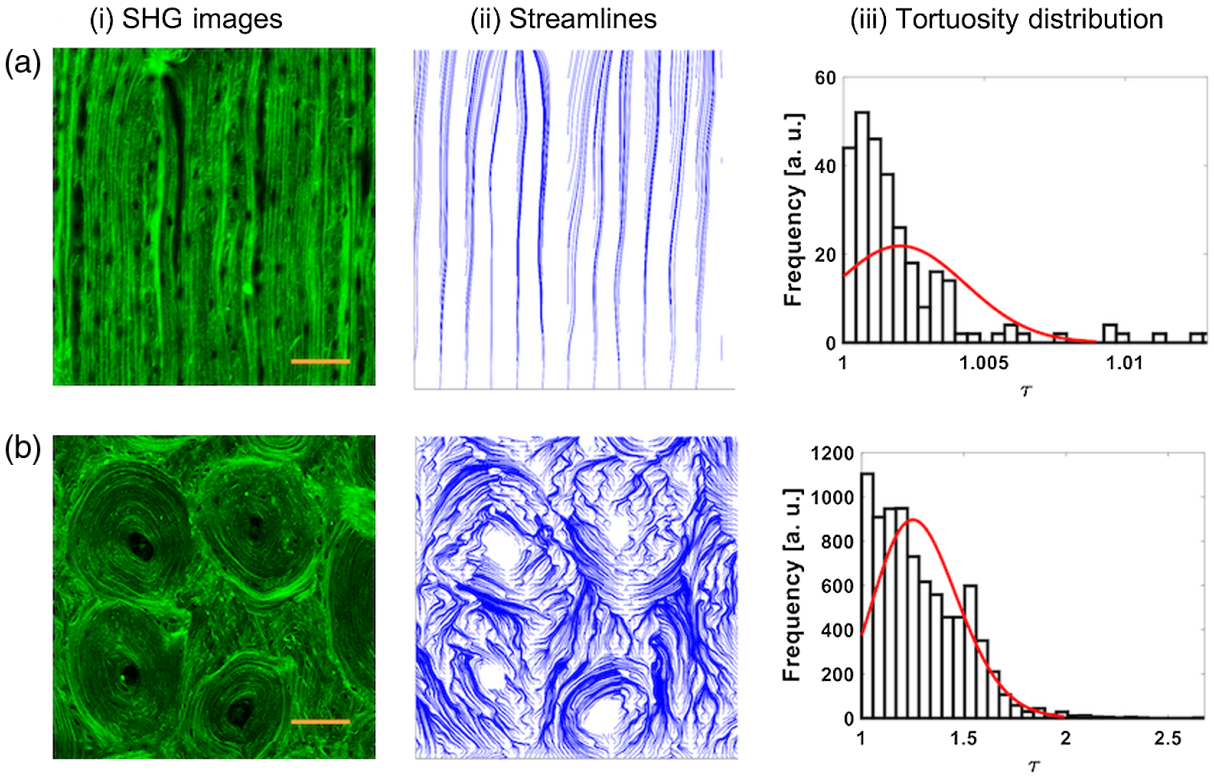

Figure 2.4: Representative SHG images of cortical bone showing (a)(i) regions with uniformly aligned fibers and (b)(i) areas containing Haversian canals. (ii) Corresponding streamline maps and (iii) tortuosity (τ) distributions are shown for each case. Scale bars indicate 200µm.

Lastly, we analyze representative SHG images of rat vaginal tissue obtained from regions near the cervix Fig. 2.5(a, i) and the introitus Fig. 2.5(b, i). Fibers in the cervical region appeared notably wavier compared to those near the introitus, where fibers exhibited more linear organization. As shown in Fig. 2.5(a, iii), the tortuosity τ for the cervical region ranges approximately from 1 to 1.5, with a peak around 1.17. The corresponding values of E and U are 1.49×10+2±46.4 and 1.37×10+2±46.35, respectively. In contrast, SHG images from the introitus region revealed a narrower τ range, spanning from 1 to 1.2 and clustering near 1.1. The values of E and U for these regions are significantly lower, measured at 4.25×10+1±11.1 and 4.37×10+1±10.4, respectively. A complete summary

of the computed parameters, including E, U, τ, and gamma distribution parameters γ_a and γ_b, is presented in Table 1.

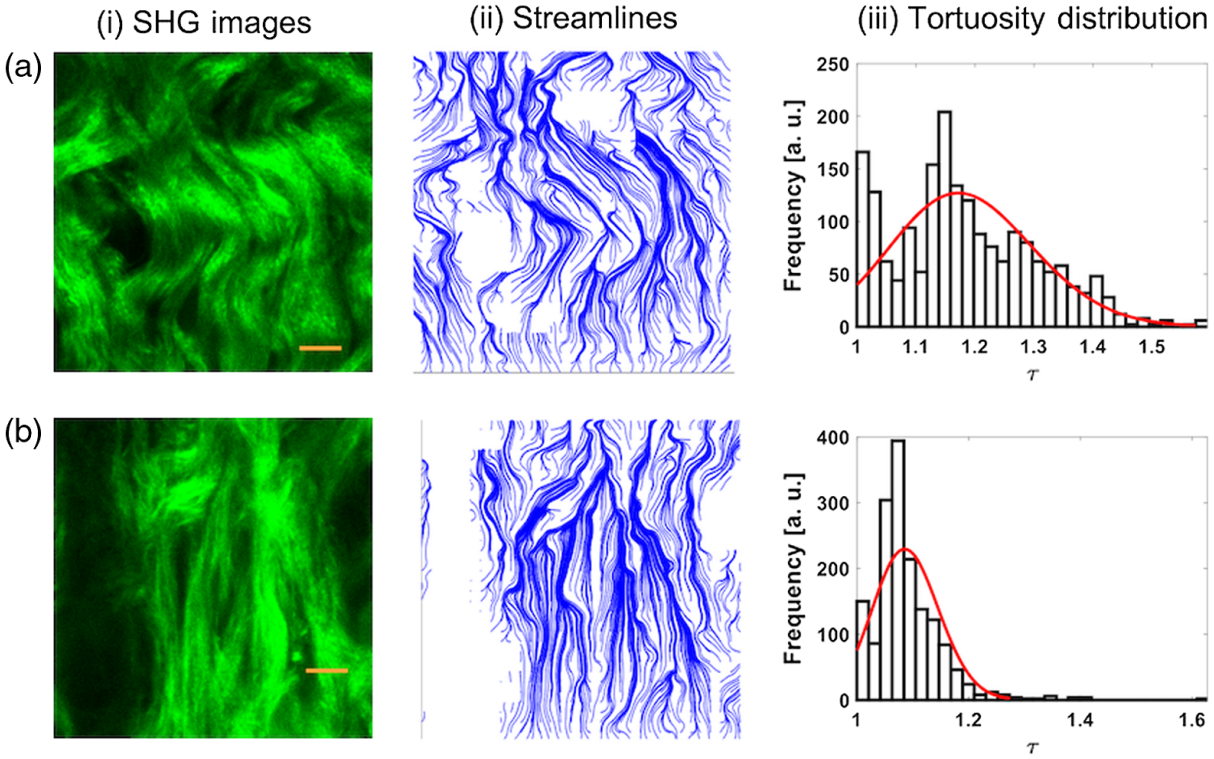

Figure 2.5: Representative SHG images of rat vaginal tissue near the (a)(i) cervix and (b)(i) introitus. (ii) Corresponding streamline maps and (iii) distributions of tortuosity (τ) values are shown for each region. Scale bars denote 10µm.

Sample	U	E	γ_a	γ_b
Straight Phantom	0	0	Inf	0
Wavy Phantom	1.92e2	1.06e2	Inf	0
Bone straight	5.70e-1±0.097	6.20e-1±0.067	6.34e5	1.58e-06
Near introitus vaginal tissue	4.25e1±11.1	4.37e1±10.4	3.40e2	3.20e-3
Near cervix vaginal tissue	1.49e2±46.4	1.37e2±46.35	9.86e1	1.20e-2
Bovine tendon	2.84e2	3.02e2	3.65e2	2.98e-3
Bone canals	2.58e3±262.7	2.64e3±270.8	3.86e1	3.33e-2

Table 2.1: Values of U and E, along with the γ-distribution parameters γ_a and γ_b derived from τ histograms, are obtained for phantom images, bovine tendon, cervical porcine bone, and rat vaginal tissue.

The samples selected for this study span a broad spectrum of collagen architectures, from predominantly straight fibers to vortex-like arrangements. We find that both E

and U correlate strongly with fiber waviness. For example, regions near the cervix of the rat vaginal tissue exhibit higher E and U values compared to those near the introitus, indicating a greater degree of waviness. A similar pattern emerges when comparing the introitus region to bovine tendon, further supporting the relationship between these metrics and fiber structure. The highest E and U values observe in this study come from Haversian canals in cortical bone, where collagen fibers exhibit vortex-like organization around the canals.

Our findings suggest that E and U can be used to classify collagen architectures into linear, wavy, and circular categories. Straight fibers typically yield values between 0 and 1, while circular fiber arrangements are associated with values starting around 80 and extending up to 3000. However, distinguishing between wavy and circular fibers becomes more difficult when E and U exceed 80, and in such cases, the inclusion ofτ is necessary for accurate classification. Although E and U tend to vary together, we anticipate that their values will diverge in volumetric datasets—such as those acquired from imaging the cornea, where fiber orientation changes with depth.

The sensitivity of our method to subtle directional changes in fiber organization is influenced by the spatial resolution of the finite difference implementation, which is limited by both optical resolution and computational constraints.

From the γ-distribution analysis of τ, we observe that γ_a is higher in tissues with uniformly aligned fibers compared to those with more heterogeneous alignment. In this analysis, each vector contributes a streamline that originates at its tail and follows the direction of the vector field. Since vectors are constrained to the upper hemisphere (0°–180°), their heads tend to orient toward the top of the image, causing streamlines to predominantly flow upward. Due to varying starting locations, this results in a higher density of streamlines at the top and lower density at the bottom of the image, an artifact that can influence both the peak and spread of the τ histogram. Despite this limitation, the method effectively distinguishes the range of collagen architectures considered in this

study.

It is important to note that the metrics developed here are derived from vector fields based on local fiber orientation and are not to be confused with similarly named metrics used in texture analysis. For instance, "energy" in texture analysis refers to grayscale homogeneity as computed through gray level co-occurrence methods[15].

2.4 Conclusion

While traditional FT-based methods can capture individual fiber orientations, they often fall short in quantifying the complexity of collagen fiber architectures. By adopting an analytical framework inspired by fluid mechanics, we are able to treat fiber orientation as part of a continuous vector field resembling fluid flow. Although this approach does not introduce new spatial information beyond what is already present in SHG images, it enables a systems-level assessment of collagen organization that reveals emergent structural patterns otherwise difficult to quantify.

Using this fluid-inspired analysis, we investigated collagen fiber organization in bovine tendon, porcine cortical bone, and rat vaginal tissue. We demonstrated that collagen orientation can be represented by pseudovectors of unit length and that the derived metrics E, U, and τ effectively differentiate between various architectural organizations. For example, tissues composed of straight fibers showed narrow τ distributions centered near 1, while increased waviness broadened the τ range. Our findings further indicate that U increases with localized directional changes, as seen in crimped or irregular fibers, and E rises with circular or vortex-like arrangements, such as those near Haversian canals. These distinctions confirm that the combined use of these metrics offers meaningful insight into the spatial organization of collagen, with implications for understanding how fiber architecture influences tissue function.

We propose that modeling fiber orientation as a vector field and applying fluid mechanics

principles offers a valuable strategy for probing the role of collagen microstructure in biological systems. This framework may enhance our understanding of tissue mechanics, particularly in the context of disease, extracellular matrix (ECM) remodeling, and aging.

References

[1] R. Ambekar, K. C. Toussaint, Jr, and A. Wagoner Johnson, "The effect of keratoconus on the structural, mechanical, and optical properties of the cornea," *J. Mech. Behav. Biomed. Mater.*, vol. 4, pp. 223–236, Apr. 2011.

[2] R. Ambekar, T.-Y. Lau, M. Walsh, R. Bhargava, and K. C. Toussaint, Jr, "Quantifying collagen structure in breast biopsies using second-harmonic generation imaging," *Biomed. Opt. Express*, vol. 3, pp. 2021–2035, Sept. 2012.

[3] J. A. McGuire, J. L. Monclova, A. C. S. Coariti, C. A. Stine, K. C. Toussaint, Jr, J. M. Munson, D. A. Dillard, and R. De Vita, "Tear propagation in vaginal tissue under inflation," *Acta Biomater.*, vol. 127, pp. 193–204, June 2021.

[4] B. Lee, J. Konen, S. Wilkinson, A. I. Marcus, and Y. Jiang, "Local alignment vectors reveal cancer cell-induced ECM fiber remodeling dynamics," *Sci. Rep.*, vol. 7, p. 39498, Jan. 2017.

[5] M. M. Kabir, V. V. Krishna Inavalli, T. Lau, and K. Toussaint, "Application of quantitative second-harmonic generation microscopy to dynamic conditions," *Biomed. Opt. Express*, vol. 4, pp. 2546–2554, Nov. 2013.

[6] T. Y. Lau, H. K. Sangha, E. K. Chien, B. L. McFarlin, A. J. Wagoner Johnson, and K. C. Toussaint, Jr, "Application of fourier transform-second-harmonic generation imaging to the rat cervix: FT-SHG OF CERVIX," *J. Microsc.*, vol. 251, pp. 77–83, July 2013.

[7] W. Lee, H. Rahman, M. E. Kersh, and K. C. Toussaint, Jr, "Application of quantitative second-harmonic generation microscopy to posterior cruciate ligament for crimp analysis studies," *J. Biomed. Opt.*, vol. 22, p. 46009, Apr. 2017.

[8] M. P. Rubbens, A. Driessen-Mol, R. A. Boerboom, M. M. J. Koppert, H. C. van Assen, B. M. TerHaar Romeny, F. P. T. Baaijens, and C. V. C. Bouten, "Quantification of

the temporal evolution of collagen orientation in mechanically conditioned engineered cardiovascular tissues," *Ann. Biomed. Eng.*, vol. 37, pp. 1263–1272, July 2009.

[9] L. Landler, G. D. Ruxton, and E. P. Malkemper, "The multivariate analysis of variance as a powerful approach for circular data," *Mov. Ecol.*, vol. 10, p. 21, Apr. 2022.

[10] M. Mazzoli, A. Molas, A. Bassolas, M. Lenormand, P. Colet, and J. J. Ramasco, "Field theory for recurrent mobility," *Nat. Commun.*, vol. 10, p. 3895, Aug. 2019.

[11] B. Jeurissen, M. Descoteaux, S. Mori, and A. Leemans, "Diffusion MRI fiber tractography of the brain," *NMR Biomed.*, vol. 32, p. e3785, Apr. 2019.

[12] I. Aranson, "The aquatic dance of bacteria," *Physics*, vol. 6, May 2013.

[13] R. A. R. Rao, M. R. Mehta, and K. C. Toussaint, Jr, "Fourier transform-second-harmonic generation imaging of biological tissues," *Opt. Express*, vol. 17, pp. 14534–14542, Aug. 2009.

[14] M. Matyka and Z. Koza, "How to calculate tortuosity easily?," AIP, 2012.

[15] Q. Zhao, C.-Z. Shi, and L.-P. Luo, "Role of the texture features of images in the diagnosis of solitary pulmonary nodules in different sizes," *Chin. J. Cancer Res.*, vol. 26, pp. 451–458, Aug. 2014.

CHAPTER 3

Fluid-Inspired Fiber Analysis of 3D Collagen Architecture

3.1 Introduction

Most of our understanding of collagen structure comes from 2D imaging approaches. While valuable, these methods cannot fully capture collagen's complex 3D architecture, a crucial element for understanding ECM remodeling and its effects on cellular processes. Additionally, experiments traditionally have used collagen in 2D cell cultures to understand how cells interact with their surroundings. However, there is an increasing need to learn from more physiologically relevant environments such as 3D collagen-based models to study cell-matrix interactions. In vivo, cells interact with a complex 3D ECM, where the process of converting physical forces into biochemical signals known as mechanotransduction differs from those in 2D environments[1]. Moreover, the 3D ECM mechanical properties regulate cell behaviors such as spreading, migration, division, stem cell differentiation, apoptosis, morphogenesis and gene expression. Cell-ECM interactions including adhesions, confinement, forces, stresses and mechanical feedback are also different in 3D compared to those in a 2D environment[2].

In recent years, there has been growing interest in the development of in vitro systems that mimic the complex three-dimensional (3D) architecture of the extracellular matrix (ECM) to better understand its role in development and disease. Various approaches have been explored, including the use of cell-laden hydrogels formulated with biological or synthetic polymers, the embedding of small numbers of cells within collagen or other bioactive materials, and the self-assembly of cell-dense microtissues. The latter involves seeding monodispersed cells in a non-adhesive environment, letting them self-assemble into a microtissue and secrete their own ECM [3–5]. Compared to other methods, self-assembly closely mimics tissue morphogenesis.

A 3D ECM not only provides structural support but also serves as a reservoir for biochemical cues, such as growth factors that influence cell behavior. Among ECM associated growth factors, transforming growth factor beta 1(TGF-β1) regulates tissue homeostasis, promotes synthesis of fibrillar collagens and remodels the ECM by increasing collagen gene expression and activating fibroblasts. Subsequently, TGF-β1 contributes to collagen- fiber density, crosslinking, and fiber alignment. These changes affect mechanical integrity, cellular responses, biochemical signaling and cell-ECM crosstalk [6, 7].

Investigating the 3D morphology of collagen in situ has become feasible thanks to z-sectioning optical imaging techniques including confocal, multiphoton and SHG microscopy. The latter has been helpful to visualize fibrotic collagen remodeling in organs such as liver, lung, heart, skin and vaginal tissue [8–12], demonstrating out- of- plane complex fiber structures that are not apparent in 2D projections alone. Computational techniques to calculate fiber orientations has facilitated the study of 3D fiber organization [8, 13, 14].

Although current methods successfully measure fiber orientation, researchers have manually delineated fibers to assess the waviness of individual collagen fibers[15]. For example, Pissarenko et al. used scanning electron microscopy images to calculate collagen curvature by overlapping circular structures to delineate fibers to evaluate fiber organization after failure[16]. While these techniques provide some insight into collagen structure in

ex vivo samples, they fall short in capturing the complexity of fiber architecture. While these techniques provide valuable insight into collagen structure in ex vivo samples, they often fall short in capturing the full complexity of fiber architecture. One way to quantify these complex structural changes is by interpreting the collagen network as a vector field where direction is orientation of the fibers. Spatial changes in the architecture of collagen fibers can be quantified by metrics inspired by fluid mechanics.

Computationally, fibrillar collagen has been simulated using pseudo vector fields (VFs) to investigate matrix dynamics involved in angiogenesis [12, 17], spontaneous cellular organization[18], remodeling processes driven by cancer cells [19, 20], tissue regeneration [21] and wound healing [22]. The 2D influence of cell properties on fiber organization has been modeled using a continuous vector field in combination with numerical simulations [21]. Vector fields have been also used to investigate the role of fibroblast-ECM interactions during scar tissue formation. In this study, fiber density defines the magnitude of the vectors, while fiber orientation determines their direction [22]. Additionally, neo vessels and collagen interactions in both 2D and 3D have been studied by modeling collagen as vector fields [12, 17]. The changes in mechanical properties due to ECM remodeling have been examined using vector fields to model collagen effects on cell self-organization [18]. For instance, cancer cells dynamic remodeling has been studied using circular statistics and vector fields to represent collagen fiber orientation [19, 20].

We previously stablished that fluid inspired fiber analysis can be applied to quantify collagen VFs in 2D ex-vivo tissue samples [23, 24]. In this chapter, we show that we can quantify 3D fibrillar architecture using fluid inspired fiber analysis (FIFA). Here, the input is a z-stack of SHG images from which we calculate fiber orientation and volume. Fiber orientation and spatial positional information is corrected using a correction algorithm to ensure fiber continuity and obtain a VF that represents the tissue structure. The VF is equivalent to the 3D velocity map of a fluid flow. Earlier used metrics such as energy (U), enstrophy (E) and tortuosity (τ) are adapted to calculate 3D fiber orientation

changes and waviness changes both in 2D and 3D. Precisely, U captures 3D positional orientation changes. E quantifies 3D circulatory architecture patterns, whereas τ computes the waviness of both in-plane and out-of-plane the fibers. To account for variations in collagen content across samples, E and U were normalized by the fiber volume yielding the parameters (E_ρ) and (U_ρ). We benchmarked our findings using computer generated volumes. Our technique has been applied to quantify fibrillar collagen structural changes during morphogenesis at days 7, 14 and 21. Our samples are cell derived ring-shaped engineered tissue constructs. Moreover, we investigated the influence of TGF-β1 to modulate the matrix at the same time points. In control samples, changes in fiber orientation show that samples become more organized by day 21. In contrast, TGF-β1 treated samples exhibit fiber disorganization by day 21. Regarding waviness, we find that for both sample groups, the τ distributions became concentrated in the lower range, indicating that the fibers became straighter by day 21. However, the range differed between control and TGF-β1 treated samples, suggesting that fibers remained wavier in the TGF-β1 groups. We find that (E_ρ), (U_ρ) and τ can successfully quantify changes in 3D collagen fiber organization from complex architectures.

3.2 Materials and methods

3.2.1 Engineered Ring-Shaped Tissues Preparation

Collagen ring- shaped tissues were engineered using normal human dermal fibroblast isolated from juvenile foreskin (PromoCell, Heidelberg, Germany). Cells were cultured in high glucose Dulbecco's Modified Eagle's medium containing phenol red, L-glutamine, and sodium pyruvate (#11995065, Thermo Fisher Scientific, Waltham, MA), supplemented with 10% fetal bovine serum (# 100-500, Gemini Bio, West Sacramento, CA) and 1% penicillin/streptomycin (#091670249, Thomas Scientific, Swedesboro, NJ). Cells were expanded through passages 4 to 8 using a standard trypsin protocol. After rinsing with

phosphate-buffered saline (PBS), the cells were detached, resuspended in fresh medium, and seeded into circular agarose molds with a 5-mm inner diameter and a surrounding 0.75-mm trough. A 50:50 optimized culture medium was used to support tissue formation, as previously described [3–5]. Within 24 hours, fibroblasts aggregated and self-organized into ring-shaped tissues. Following this initial period, some rings were treated with 10 ng/mL TGF-β1 for the remainder of the experiment, while others remained untreated as controls. Culture medium was refreshed every 2–3 days, and tissues were maintained in a humidified incubator at 37°C with 10% CO_2. The resulting tissues had an approximate thickness of 250 μm.

3.2.2 Image Acquisition

Before imaging, tissue rings were fixed in 10% formalin while still embedded in their agarose molds. The samples were then stored at 4°C, rinsed with PBS, and placed in a petri dish for imaging. Multiphoton imaging was performed using an Olympus FV-1000-MPE microscope equipped with a Mai Tai HP tunable laser, spectrally centered at 790nm. SHG signals were captured using a 25x water-immersion objective (Olympus XLPlan N, N.A. 1.05, WD 2mm) and filtered through a 405/40 bandpass cube.

3.2.3 Fiber quantification

A custom- built MATLAB (MathWorks, Natick, MA) script developed in our lab was used to compute the 3D orientation of collagen fibers from a z-stack of images [10, 25]. The volumetric dataset was initially partitioned into a 3D grid to enable localized analysis of fiber bundle orientation. Fiber boundaries were then identified using the Canny edge detection algorithm and within each grid cell, we applied a filter bank approach that compared the 3D Fourier transform of the image data to a set of predefined 3D orientation filters in Fourier space. The orientation corresponding to the highest correlation is selected

as the fiber's local orientation. Each result was classified as dark, one-directional, two-directional, or isotropic. The preferred orientation is expressed in two angular components θ and ϕ, representing the in plane(x-y) and out-of-plane (z-) fiber directions, respectively[8].

Our preliminary fiber orientation calculations bound θ from 0° to 360°, and ϕ between 0° to 90°. The limit of ϕ excludes vectors that are colinear but point in opposite directions, which are essential for representing crimped fibers and preserving fiber continuity. To address this, we developed a direction-correction algorithm. The procedure started by generating a binary mask to identify isotropic and dark regions. Next, a network of dots and lines is created by connecting each vector to its neighboring vectors in 3D space, including diagonal neighbors. The binary mask ensures that edges were not formed through dark or isotropic regions of interests (ROIs). Finally, the correction algorithm chooses the vector corresponding to the angle θ as defined in (Eq. 3.1).

$$\theta = min \left\{ \frac{ab}{|a||b|}, \frac{a_{nt}b}{|a_{nt}||b|} \right\}. \tag{3.1}$$

Here, the vector that needs to be corrected was a, its antiparallel counterpart was a_{nt}, and b was a neighboring vector. To summarize, we traversed the entire graph using a breath-first search [26], computed the angular distance between each neighboring vector on a unit sphere represented in Cartesian coordinates. Each neighboring vector's orientation was also rotated by 180 in both θ and ϕ to re-evaluate the distance. The algorithm outputted the angle configuration corresponding to the shortest calculated distance.

A bivariate tiled histogram was constructed to examine the joint distribution of θ and ϕ. The data were binned using uniform 15° intervals along both axes. Frequencies were visualized using a modified plasma color scale, where bins with zero frequency were shown in gray (RGB: 142, 142, 142). Angular distributions of θ and ϕ were also represented using polar plots. In these plots, the angular axis denoted degrees, while the radial axis indicated the frequency of occurrences. The angular axis was segmented into 15° intervals,

and the radial axis displayed normalized frequencies. All visualizations were generated using MATLAB.

Fluid-derived metrics including τ, (U_ρ), and (E_ρ) were calculated using their respective formulas [27, 28]. To compare the proportional distributions of (U_ρ), and (E_ρ), a double doughnut chart was generated, with (U_ρ)represented in the outer ring and (E_ρ) in the inner ring. Values were expressed as percentages relative to the maximum within each sample group. This chart was produced using OriginPro (OriginLab Corp., Northampton, MA), and labels were added to indicate the proportion of each category. To compute τ, streamlines tangential to the vector field (VF) were generated from each vector. The length (L) of each streamline was measured and divided by the distance (D) between its endpoints, yielding τ (as defined in Eq. 3.2). The distribution of τ values was visualized in a histogram to illustrate waviness frequency across the volume. The following equations were used to calculate τ, (U_ρ), and (E_ρ) in 3D:

$$\tau = \frac{L}{D}, \tag{3.2}$$

$$E_\rho = \frac{\int_V |\nabla \times \vec{v}|^2 dV}{V}, \tag{3.3}$$

and

$$U_\rho = \frac{\int_V |\nabla \cdot \vec{v}|^2 dV}{V}, \tag{3.4}$$

In this context, ∇ denoted the Del operator, $\vec{v}$ represented the VF describing the 3D orientation of collagen fibers, V referred to the volume of the z-stack, andρ indicated the fluid density, which was simplified to one, as it served as a proportional constant across all samples. The collagen fiber volume (V) was determined by first binarizing the SHG data to isolate fiber regions. This binary mask was then multiplied by the z-stack interval to calculate each slice's volumetric contribution. The total fiber volume was

obtained by summing these values across all slices. To compute the integrals of E and U, a 3D forward finite difference method was applied, transforming the continuous equations into discrete sums. Each result was then normalized by the total volume (V) to derive the densities ((E_ρ) and (U_ρ)). A visual summary of this procedure was provided in Fig. 3.1.

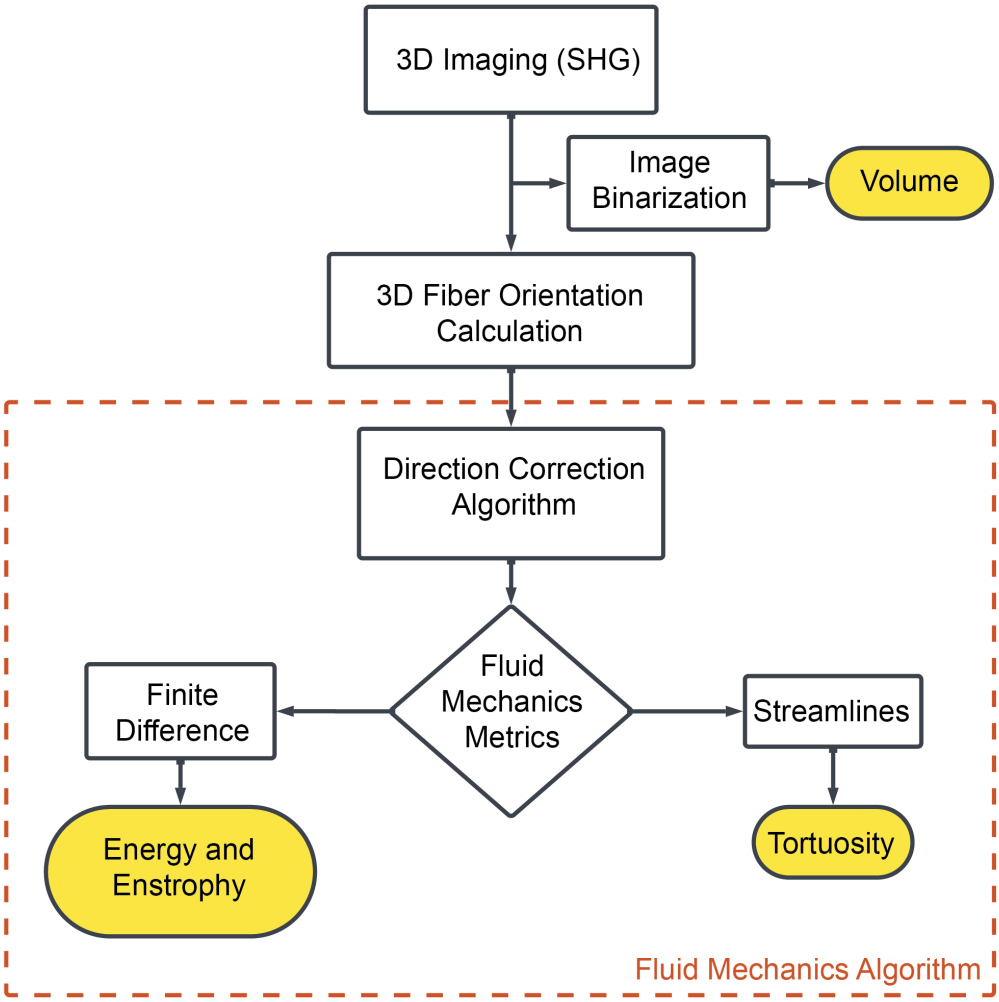

Figure 3.1: Flow diagram of the algorithm used to convert image data to a 3D VF and the accompanying calculated fluid mechanics metrics

3.3 Results

3.3.1 Analysis of Calibration Data Using FIFA algorithm

Figure 3.2A shows representative volumetric reconstructions (green) overlaid with a VF (purple) that indicated fiber orientation. The synthetic volumes are measured in micrometers. In the straight fiber volume, the vectors are mostly aligned in a single direction, whereas the undulated fiber volume demonstrate a wider range of vector orientations. Figure 3.2B displays a bivariate tiled histogram representing the joint distribution of the in-plane and out-of-plane angles, θ and ϕ, respectively. Color intensity reflected the frequency of observations within the volume, facilitating the identification of density patterns and relationships between the two orientation angles. In the straight fiber volume, the histogram shows a concentrated distribution at $\theta = 90^\circ$ and $\phi = 30^\circ$, consistent with the volumetric reconstruction. In contrast, the undulated fiber volume reveals a cluster of data points in the lower-left quadrant, with a preferred orientation of $\theta = 90^\circ$ and $\phi = 110^\circ$. Figure 3.2C illustrates a polar plot of the θ and ϕ distributions as a function of angular position. The radial axis denotes the frequency of occurrence, while the angular axis corresponds to in-plane orientation. This visualization helps to identify dominant fiber orientations both in-plane and out-of-plane. In the straight fiber volume, the polar plot for θ shows a dominant orientation at $\theta = 90^\circ$, and the polar plot for ϕ indicates a preferred angle of 30°, aligning with the results shown in the bivariate histogram. In the undulated fiber volume, the polar plot of θ indicates a preferred orientation of 90°, while the polar plot of ϕ reveals two distinct peaks at 75° and 110°.

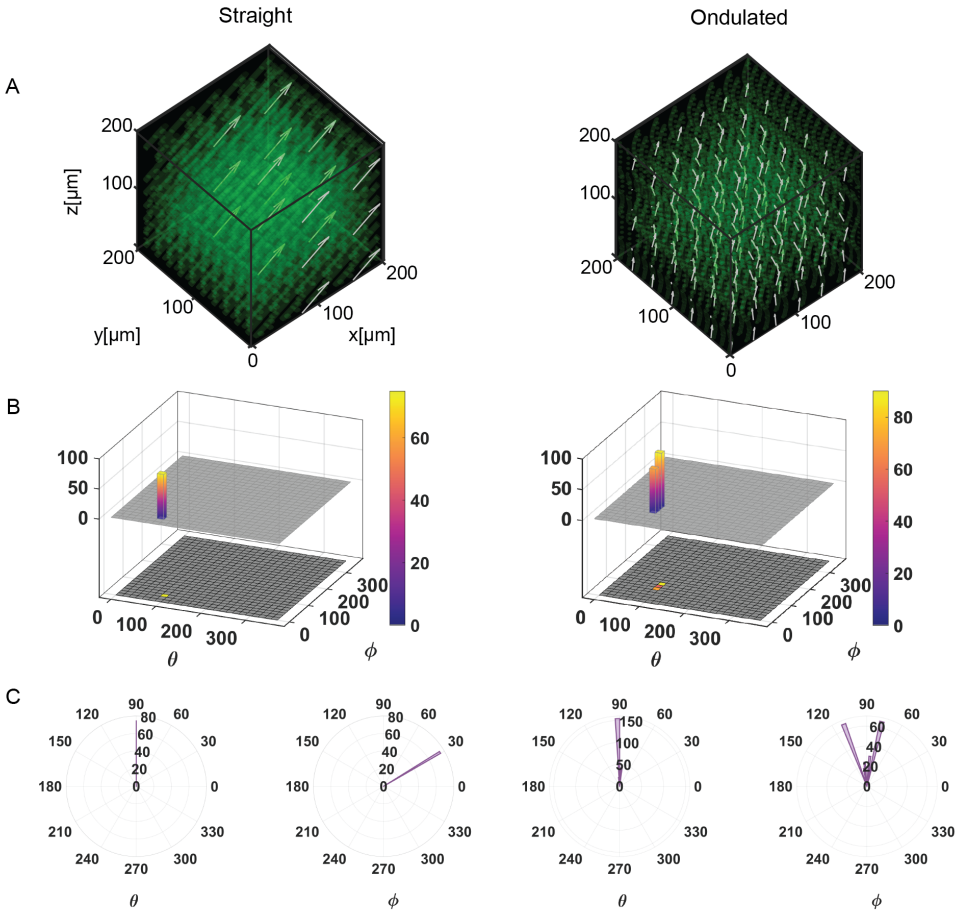

Figure 3.2: (A) Representative rendered straight and undulated phantom fiber volumes with overlaid 3D VFs. (B) Bivariate tiled histogram of in-plane and out-of-plane orientations. (C) Polar plots indicating orientation frequency of θ and ϕ angles throughout the volume

Figure 3.3 illustrates the FIFA metrics: E_ρ, U_ρ, and τ. In Figure 3.3A, volumetric reconstructions (green) are shown alongside streamlines colored using the parula palette. These streamlines are derived from the fiber orientation vector fields (VFs), with each one initiating from a vector. The streamline colors indicate different starting positions relative to the z-axis. In the straight fiber sample, the streamlines align uniformly in one direction, whereas in the undulated fiber sample, the streamlines reveal the crimped fiber architecture. Figure 3.3B depicts the relative percentages of E_ρ and U_ρ, which capture the degree of orientation change. As expected, the straight fiber volume exhibits no significant variation in fiber direction, yielding 0% relative values for both E_ρ and U_ρ when compared

to the undulated sample, which is normalized to 100%. Figure 3.3C presents histogram distributions of τ values derived from the streamlines. For the straight fiber volume, τ values are tightly centered around 1, consistent with linear streamline geometry. In contrast, the undulated sample shows τ values ranging from 1 to 1.038, with distinct frequency peaks that reflects variations in wave crest numbers associated with different z-axis starting points.

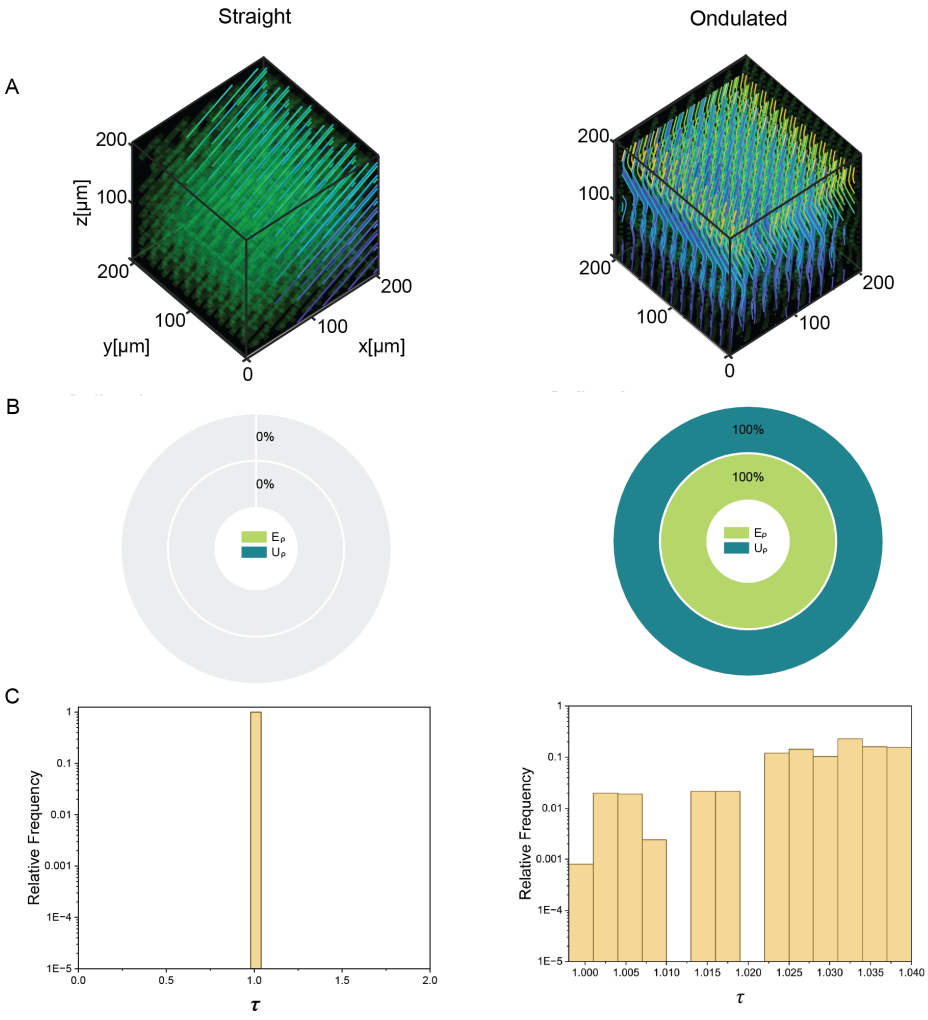

Figure 3.3: Calculated fluid-inspired metrics: (A) Streamlines, (B) (E_ρ) (light green) and (U_ρ) (dark green), and (C) Tortuosity values for phantom straight and undulated fibers

3.3.2 Fiber architecture changes in ring tissues over three developmental stages

Figure 3.4A shows SHG images taken at the midpoint of the z-stack for control ring tissues collected on Days 7, 14, and 21. The corresponding volumetric reconstructions, overlaid with VFs indicating fiber orientation, are presented in Figure 3.4B. Figure 3.4C depicts bivariate histograms illustrating the joint distributions of θ and ϕ angles across these time points. A total of 17,496 vectors are used to characterize the fiber architecture per biological sample across each time point. On Day 7, the dominant orientation is centered at θ between 295°–305° and ϕ between 195°–205°, with a peak frequency of 9.64%. This is accompanied by smaller clusters and a widespread distribution of low-frequency, disorganized fiber orientations. By Day 14, the preferred angular region is shifted to θ between 205°–215° and ϕ between 345°–355°, with a frequency of 8.36%. On Day 21, the orientation distribution shows the highest peak frequency of 11.7%, with θ between 115°–125° and ϕ between 165°–175°. Figure 3.4D presents polar plots of fiber orientation distributions over time. On Day 7, the most frequent θ value occurred between 120°–130° (8.77%), while the dominant ϕ orientation is between 10°–20° (16.64%). By Day 14, the primary θ peak shifts to 200°–210° (9.5%), and the most frequent ϕ value move to 340°–350° (8.46%). On Day 21, the preferred θ and ϕ ranges are 110°–120° (8.96%) and 160°–170° (12.98%), respectively.

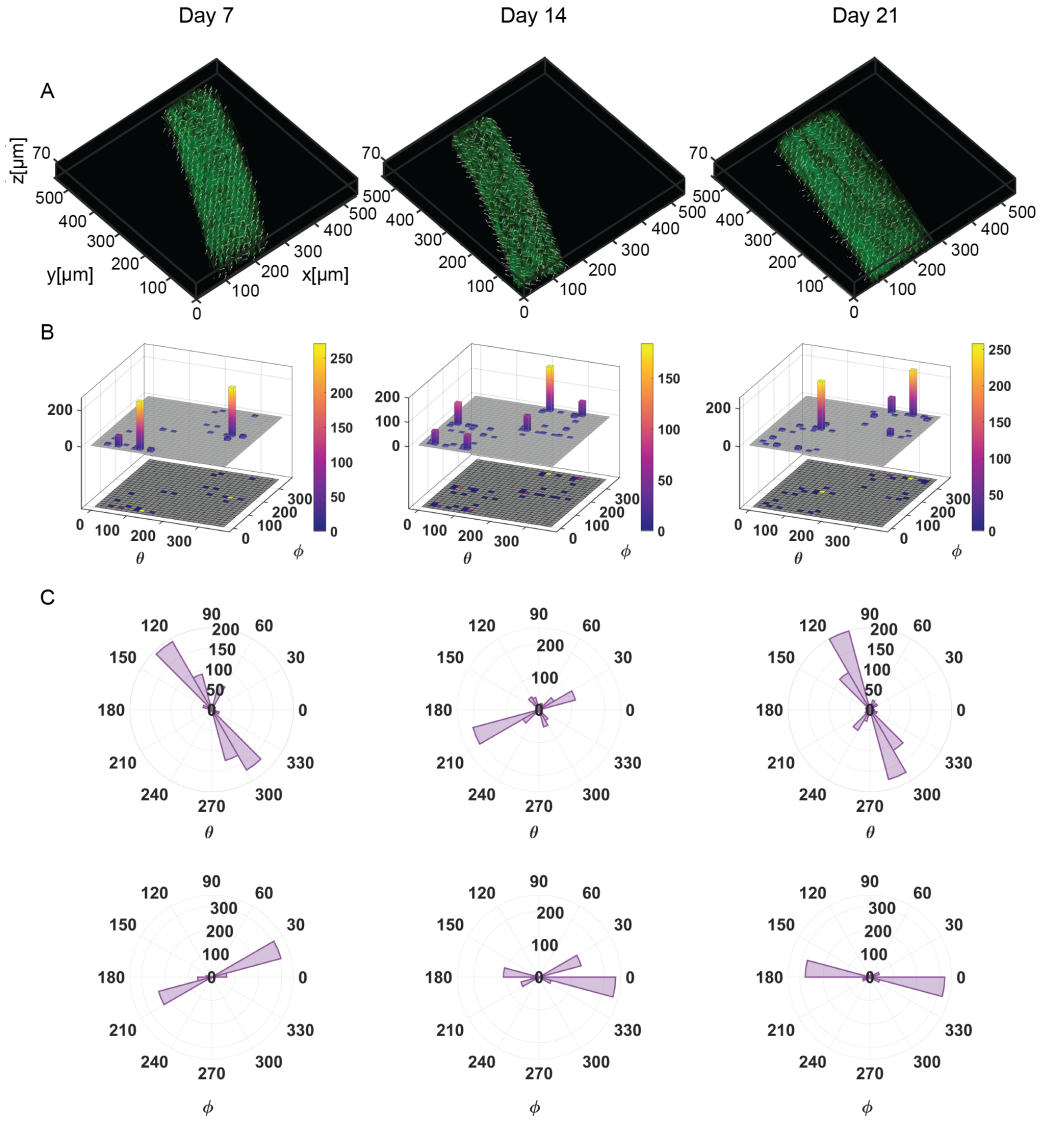

Figure 3.4: (A) Representative SHG images showing engineered 3D ring-shaped tissues over 21 days. (B) Volumetric SHG rendered images overlaid with computed 3D VF of collagen fiber orientations. (C) Bivariate tiled histogram of in-plane and out-of-plane orientations. (D) Polar plots indicating orientation frequency of θ and ϕ angles throughout the volume

Figure 3.5A shows streamlines derived from fiber orientation VFs at the same time points. Figure 3.5B summarizes the relative changes in E_ρ and U_ρ, with values compared to the most disorganized condition Day 21 with TGF-β1. On Day 7, E_ρ and U_ρ are 29.9% and 30.4%, respectively. These values increase by Day 14 to 67.6% (E_ρ) and 73.7% (U_ρ), indicating enhanced fiber organization. By Day 21, E_ρ decreases to 43.7%, and

U_ρ drops to 53.1%, suggesting partial reorganization or degradation of alignment. The orientation-related metric τ also shows time-dependent variation. On Day 7, τ values are narrowly distributed around 1, with a histogram range extending up to 2000. By Day 14, the frequency of τ near 1 decreases, and the range broadens to 2300, reflecting increased orientation complexity. On Day 21, τ values decline in frequency around 1 compared to Day 7, and the overall range narrowed again to between 1 and 2000.

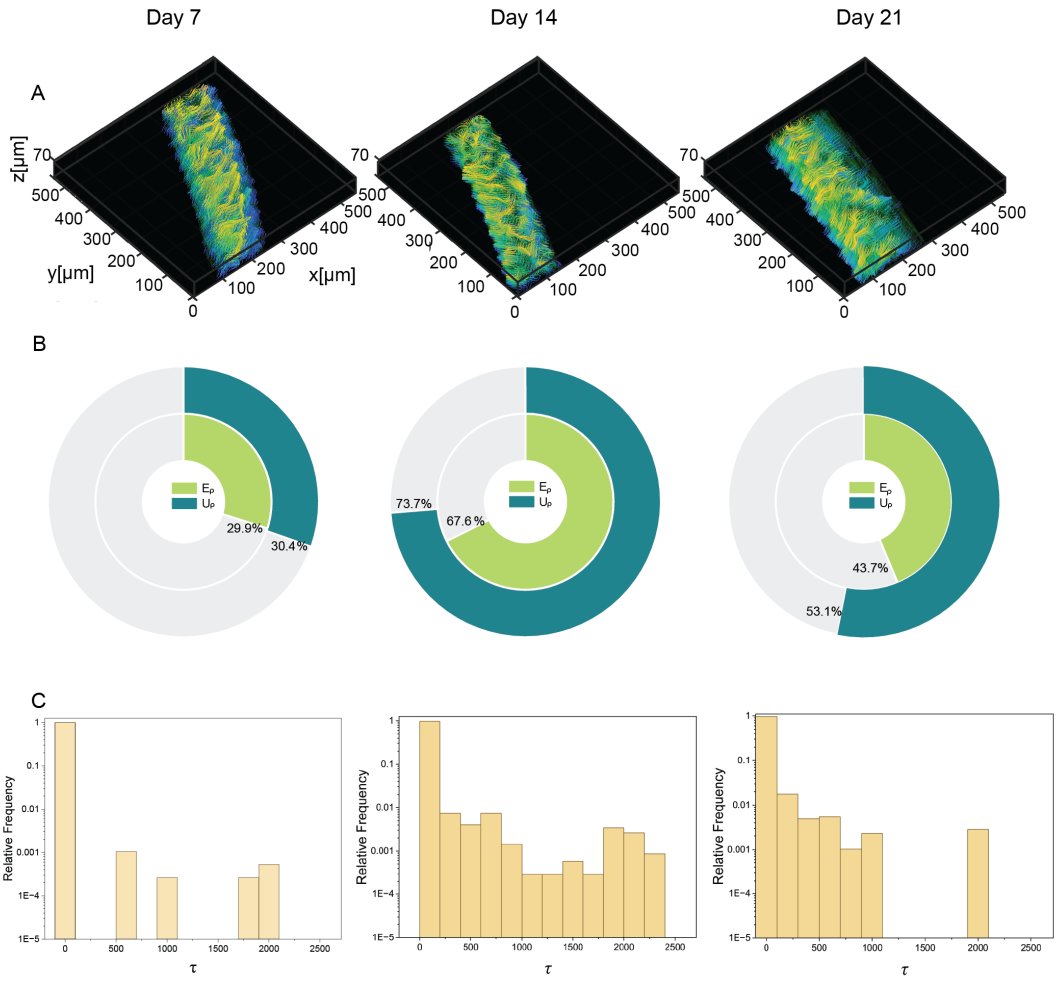

Figure 3.5: Calculated fluid-inspired metrics over 21 days of ring tissue development: (A) Streamlines, (B) (E_ρ) (light green) and (U_ρ) (dark green), and (C) Tortuosity values highlighting architectural changes

3.3.3 TGF-β1 influence on ring- tissue formation across three developmental stages

Figure 3.6A presents SHG images taken at the midpoint of the z-stack for control rings on Days 7, 14, and 21. The corresponding volumetric reconstructions, overlaid with VFs representing fiber orientation, are shown in Figure 3.6B. In Figure 3.6C, the bivariate histogram illustrates the joint distribution of θ and ϕ across time points. On Day 7, the preferred orientation is located at the intersection of θ (25°–35°) and ϕ (165°–175°), with a frequency of 3.43%, alongside smaller clusters and scattered random orientations. By Day 14, the preferred region shifts to θ (115°–125°) and ϕ (25°–15°), with a frequency of 7.45%. On Day 21, the most frequent orientation appears at θ (155°–165°) and ϕ (10°–20°), though at a reduced frequency of 1.18%. Figure 3.6D shows polar plots depicting the temporal changes in fiber orientation. On Day 7, θ values were concentrated between 20°–30% (4.5%), and ϕ values peak between 160°–170° (6.99%). On Day 14, the dominant θ range shifts to 120°–130° (8.14%), and ϕ peaks at 10°–20° (13.44%). On Day 21, the preferred θ values occur between 60°–70° (3.20%), while ϕ peaks between 10°–20° (3.52%).

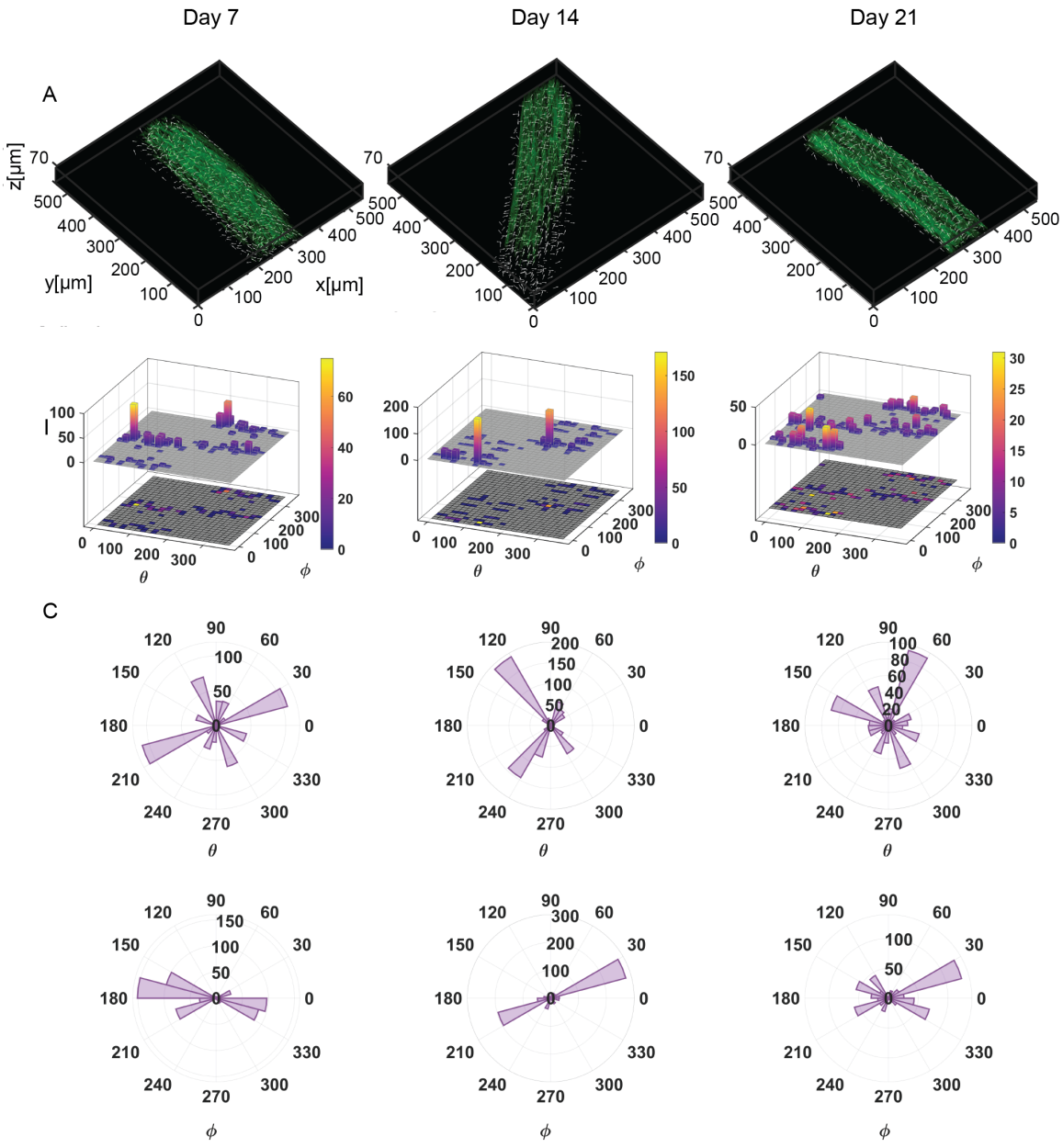

Figure 3.6: (A) Representative SHG images showing the developmental effects of TGF-β1 on ring tissues over 21 days. (B) Collagen I fiber orientation VF from 3D rendered SHG images. (C) Bivariate tiled histogram of 3D vector directions. (D) Polar plots showing the frequency distribution of θ and ϕ angles throughout the volume

Figure 3.7A displays streamlines generated from fiber orientation VFs in TGF-β1-treated rings at Days 7, 14, and 21. Figure 3.7B illustrates the relative changes in E_ρ and U_ρ. On Day 7, E_ρ and U_ρ begin at 75.8% and 88.1%, respectively. By Day 14, E_ρ increases

slightly to 76.5%, and U_ρ reached 83.9%. On Day 21, both metrics peaked at 100%. Figure 3.7C presents τ histograms capturing the distribution of streamline alignment over time. On Day 7, τ values range from 0 to 2750. By Day 14, the distribution narrows, ranging from 0 to 2300. On Day 21, the τ range extends slightly from 0 to 2710.

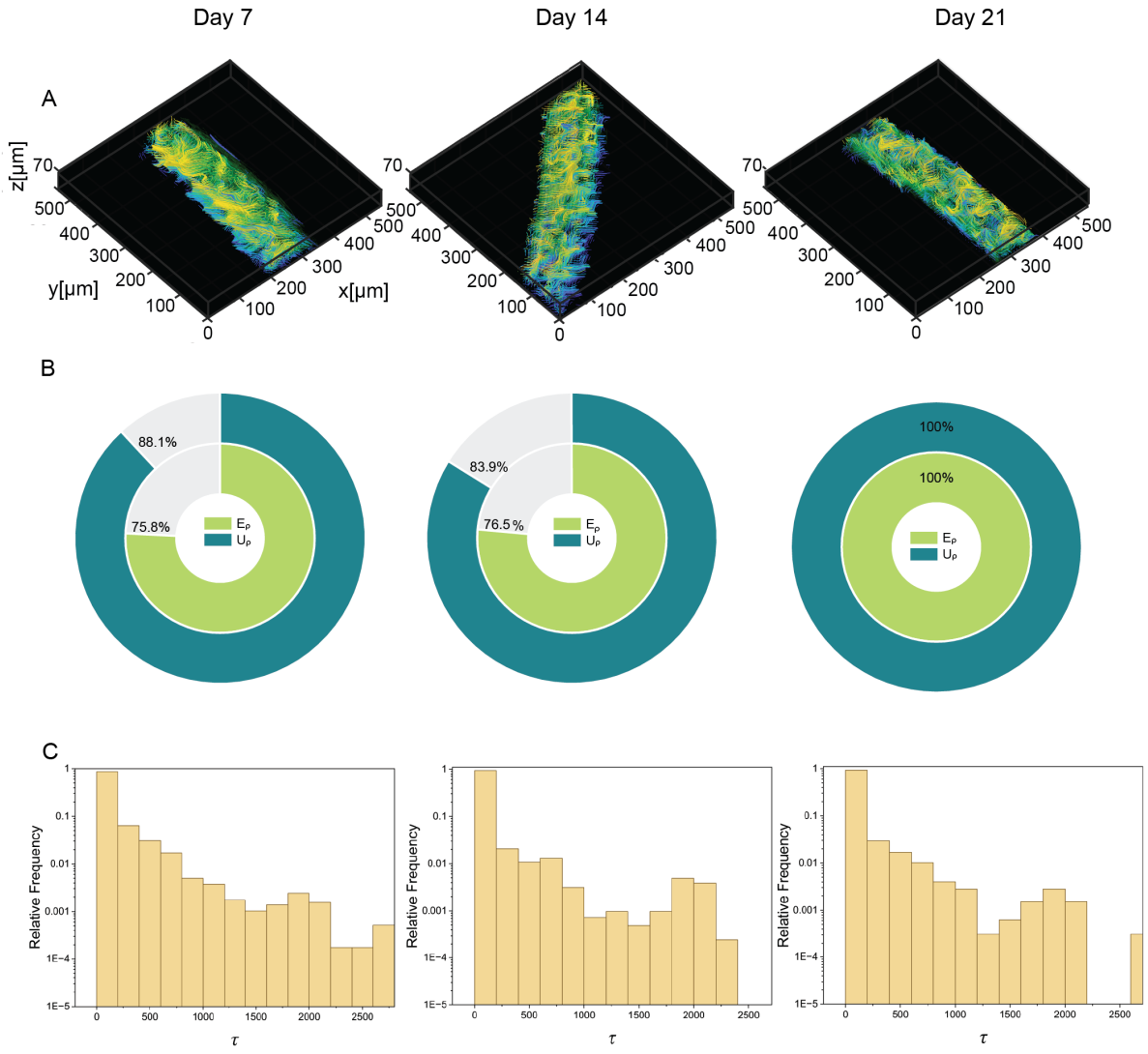

Figure 3.7: Calculated fluid-inspired metrics over 21 days in the presence of TGF-β1 of ring tissue development: (A) Streamlines, (B) (E_ρ) (light green) and (U_ρ) (dark green), and (C) Tortuosity values highlighting architectural changes

Sample	Type	U_ρ	E_ρ
Straight	Phantom	0	0
Ondulated	Phantom	1.38E-05	7.02E-06
Day 7	Control	3.66E-05	2.74E-05
Day 14	Control	8.58E-05	6.64E-05
Day21	Control	5.34E-05	4.78E-05
Day 7	TGF-β1	9.28E-05	7.94E-05
Day 14	TGF-β1	9.36E-05	7.56E-05
Day21	TGF-β1	1.22E-04	9.01E-05

Table 3.1: Summary of E_ρ and U_ρ calculations for phantom and ring-shaped tissues

3.4 Discussions

Our findings showed that both bivariate and polar plots produced consistent results for homogeneous fibrous samples, confirming that each visualization method was equally effective in capturing uniform fiber orientation. However, in samples where orientation varied across the volume, polar plots reflected the dominant global orientation but failed to represent localized frequency variations as effectively as bivariate histograms. This distinction indicated that while polar plots were useful for summarizing overall alignment patterns, bivariate histograms offered a more detailed view of preferred orientation. Additionally, bivariate histograms exhibited clustered value distributions in cases of gradual fiber orientation changes, as seen in our analysis of undulated phantom volumes. This suggested that at sub-micrometer resolution, our approach successfully captured crimped fibers along the z-axis, which was especially relevant for out-of-plane orientation. The new metrics introduced E_ρ, U_ρ and τ provided deeper insight into spatial changes in fiber orientation density and waviness. E_ρ, captured rotational local fiber changes, while U_ρ quantified broader orientation variations. For linearly aligned fibers, both E_ρ, and U_ρ equaled zero, indicating a constant orientation along all axes; thus, lower E_ρ, and U_ρ values were associated with homogeneous samples. As presented in Table 1, the presence of undulations led to higher E_ρ, values relative to U_ρ, suggesting that curvature contributed more significantly to rotational gradients. Our waviness analysis

further demonstrated that τ equaled 1 for perfectly straight fibers, while undulated fibers displayed distributions with distinct peaks. The peak at $\tau = 1$ likely stemmed from edge artifacts or analytical constraints, such as vector origins near z-stack boundaries, emphasizing the importance of careful selection of regions to minimize boundary effects. Overall, results from applying FIFA to investigate architectural changes in engineered ring tissues revealed that in roughly one-third of samples, the preferred θ and ϕ from polar plots did not match the bivariate distributions. In the remaining samples, frequencies in the bivariate plots were consistently lower than those shown in polar plots. These findings underscored the importance of 3D analysis for evaluating fiber organization over time. In Figure 3.2, the Day 14 control sample exhibited greater ϕ variance, indicating that early remodeling involved significant out-of-plane reorientation. This highlighted a key limitation of 2D analyses, which risked misrepresenting fiber architecture by overlooking structural disorganization. By Day 21, fiber alignment improved, as seen by dominant preferred orientations in polar plots and a decrease in competing orientations. This progression suggested that fiber organization became more ordered over time, likely reflecting biological processes such as ECM remodeling or mechanical stabilization. These results aligned with previous findings that structural organization increases with tissue maturation. The FIFA metrics supported these trends (Fig. 3.5), showing the highest E_ρ value at Day 14, which implied that fiber reorganization was most active at this stage, potentially due to remodeling processes. Additionally, τ values covered the broadest range at Day 14, indicating heightened structural heterogeneity as fibers transitioned from disordered to aligned states. At Day 7, τ values were most frequently around 1, reflecting a greater prevalence of straight fibers. By Day 21, the τ distribution narrowed and fewer values clustered around 1, pointing to a more stabilized architecture, possibly driven by ongoing matrix remodeling and fiber realignment. These findings reinforced the value of tracking fiber organization as a dynamic maturation process from disordered to structured states. The presence of TGF-β1 significantly influenced fiber organization over time (Fig. 3.6). At Day 7, θ disorganization increased, suggesting that TGF-β1

initially disrupted alignment. By Day 14, ϕ variance decreased, implying limited structural reorganization without clear directional alignment along the z-axis. Notably, the highest frequency of preferred orientation occurred at this point, potentially marking a transient state of partial organization. By Day 21, fibers appeared disorganized in both in-plane and out-of-plane directions, suggesting that TGF-β1 ultimately resulted in a less ordered network. This disorganization may have arisen from excessive remodeling, unbalanced collagen deposition/degradation, or cell-mediated processes that hindered stabilization. These findings suggested that although TGF-β1 played a major role in remodeling, its impact on fiber organization varied with tissue maturation. The FIFA metrics further revealed a time-dependent decrease in fiber waviness (Fig. 3.7), resulting in more compact tissue structure and reduced volume. Although τ values appeared to rise by Day 21, most values still skewed toward the lower end of the distribution, as indicated by a left-skewed histogram. This implied that in the presence of TGF-β1, as the matrix matured, fibers straightened and packed more densely. Such compaction may have reflected increased matrix remodeling, a crucial factor in tissue development and function. Interestingly, although straightening typically suggested better organization, Day 21 samples displayed the highest disorganization per unit volume. Despite the strengths of our method, some limitations were acknowledged. FIFA's resolution depended on the spatial resolution of the microscopy technique used to collect volumetric datasets. Boundary regions introduced artifacts, often reflected as τ values around 1. Currently, $\tau = 1$ values were filtered out as artifacts, though future work could develop a correction algorithm to define this threshold based on sample characteristics. Additionally, streamline generation could be improved by adopting a more representative distribution of starting points that better captured fiber density. Refining these aspects would enhance the accuracy of collagen fiber mapping and offer a more faithful representation of fiber architecture. Addressing these limitations in future work would strengthen the methodology and improve fiber analysis in complex tissues.

3.5 Conclusion

In this chapter, we presented FIFA, a novel technique for quantifying 3D collagen fiber architecture. Integrating SHG imaging with Fourier analysis and metrics from fluid mechanics, FIFA enabled volumetric analysis of fiber orientation in 3D (θ,ϕ), spatial orientation change density (E_ρ), rotational gradients (U_ρ), and streamline waviness (τ). This method introduces new metrics for characterizing fiber geometry and structural dynamics, going beyond conventional orientation analysis traditionally used for collagen assessment. We applied FIFA to collagen-rich, ring-shaped tissues at various maturation stages to investigate how TGF-β1 affects fiber organization. Our results showed that TGF-β1 treatment induces increased fiber disorganization relative to controls, supporting its known role in extracellular matrix (ECM) remodeling. FIFA offers a powerful tool for exploring tissue architecture and maturation, with potential applications in diagnosing and monitoring diseases marked by ECM alterations. Given the widespread use of collagen-based biomaterials in tissue engineering, this method provides a robust platform for evaluating fiber organization in engineered constructs.

References

[1] A. Saraswathibhatla, D. Indana, and O. Chaudhuri, "Cell-extracellular matrix mechanotransduction in 3D," *Nat. Rev. Mol. Cell Biol.*, vol. 24, pp. 495–516, July 2023.

[2] B. M. Baker and C. S. Chen, "Deconstructing the third dimension: how 3D culture microenvironments alter cellular cues," *J. Cell Sci.*, vol. 125, pp. 3015–3024, July 2012.

[3] B. T. Wilks, E. B. Evans, M. N. Nakhla, and J. R. Morgan, "Directing fibroblast self-assembly to fabricate highly-aligned, collagen-rich matrices," *Acta Biomater.*, vol. 81, pp. 70–79, Nov. 2018.

[4] C. M. Hopkins, B. T. Wilks, and J. R. Morgan, "TGF-1 requires IL-13 to sustain collagen accumulation and increasing tissue strength and stiffness," *Connect. Tissue Res.*, vol. 66, pp. 107–120, Mar. 2025.

[5] B. T. Wilks, E. B. Evans, A. Howes, C. M. Hopkins, M. N. Nakhla, G. Williams, and J. R. Morgan, "Quantifying cell-derived changes in collagen synthesis, alignment, and mechanics in a 3D connective tissue model," *Adv. Sci.*, vol. 9, p. e2103939, Apr. 2022.

[6] B. Hinz, "The extracellular matrix and transforming growth factor-1: Tale of a strained relationship," *Matrix Biol.*, vol. 47, pp. 54–65, Sept. 2015.

[7] K. K. Kim, D. Sheppard, and H. A. Chapman, "TGF-1 signaling and tissue fibrosis," *Cold Spring Harb. Perspect. Biol.*, vol. 10, p. a022293, Apr. 2018.

[8] W. Lee, A. O. Moghaddam, Z. Lin, B. L. McFarlin, A. J. Wagoner Johnson, and K. C. Toussaint, "Quantitative classification of 3D collagen fiber organization from volumetric images," *IEEE Trans. Med. Imaging*, vol. 39, pp. 4425–4435, Dec. 2020.

[9] T. Y. Lau, R. Ambekar, and K. C. Toussaint, "Quantification of collagen fiber

organization using three-dimensional fourier transform-second-harmonic generation imaging," *Opt. Express*, vol. 20, pp. 21821–21832, Sept. 2012.

[10] W. Lee, A. O. Moghaddam, B. L. McFarlin, A. J. Wagoner Johnson, and K. C. Toussaint, "Analysis of 3D collagen organization in non-pregnant rat cervix tissue," p. JW3A. 92, Sept. 2019.

[11] T.-Y. Lau, R. Ambekar, and K. C. Toussaint, "Three-dimensional fourier-transform second-harmonic generation microscopy for quantification of collagen fiber organization in biological tissues," in *Frontiers in Optics 2012/Laser Science XXVIII*.

[12] S. A. LaBelle, A. M. Poulson, 4th, S. A. Maas, A. Rauff, G. A. Ateshian, and J. A. Weiss, "Spatial configurations of 3D extracellular matrix collagen density and anisotropy simultaneously guide angiogenesis," *PLoS Comput. Biol.*, vol. 19, p. e1011553, Oct. 2023.

[13] Z. Liu, L. Speroni, K. P. Quinn, C. Alonzo, D. Pouli, Y. Zhang, E. Stuntz, C. Sonnenschein, A. M. Soto, and I. Georgakoudi, "3D organizational mapping of collagen fibers elucidates matrix remodeling in a hormone-sensitive 3D breast tissue model," *Biomaterials*, vol. 179, pp. 96–108, Oct. 2018.

[14] M. Sun, A. B. Bloom, and M. H. Zaman, "Rapid quantification of 3D collagen fiber alignment and fiber intersection correlations with high sensitivity," *PLoS One*, vol. 10, p. e0131814, July 2015.

[15] R. Rezakhaniha, A. Agianniotis, J. T. C. Schrauwen, A. Griffa, D. Sage, C. V. C. Bouten, F. N. van de Vosse, M. Unser, and N. Stergiopulos, "Experimental investigation of collagen waviness and orientation in the arterial adventitia using confocal laser scanning microscopy," *Biomech. Model. Mechanobiol.*, vol. 11, pp. 461–473, Mar. 2012.

[16] A. Pissarenko, W. Yang, H. Quan, K. A. Brown, A. Williams, W. G. Proud, and

M. A. Meyers, "Tensile behavior and structural characterization of pig dermis," *Acta Biomater.*, vol. 86, pp. 77–95, Mar. 2019.

[17] L. T. Edgar, S. A. Maas, J. E. Guilkey, and J. A. Weiss, "A coupled model of neovessel growth and matrix mechanics describes and predicts angiogenesis in vitro," *Biomech. Model. Mechanobiol.*, vol. 14, pp. 767–782, Aug. 2015.

[18] S. Checa, M. K. Rausch, A. Petersen, E. Kuhl, and G. N. Duda, "The emergence of extracellular matrix mechanics and cell traction forces as important regulators of cellular self-organization," *Biomech. Model. Mechanobiol.*, vol. 14, pp. 1–13, Jan. 2015.

[19] B. Lee, J. Konen, S. Wilkinson, A. I. Marcus, and Y. Jiang, "Local alignment vectors reveal cancer cell-induced ECM fiber remodeling dynamics," *Sci. Rep.*, vol. 7, p. 39498, Jan. 2017.

[20] S. Poonja, A. Forero Pinto, M. C. Lloyd, M. Damaghi, and K. A. Rejniak, "Dynamics of fibril collagen remodeling by tumor cells: A model of tumor-associated collagen signatures," *Cells*, vol. 12, Nov. 2023.

[21] J. C. Dallon, J. A. Sherratt, and P. K. Maini, "Mathematical modelling of extracellular matrix dynamics using discrete cells: fiber orientation and tissue regeneration," *J. Theor. Biol.*, vol. 199, pp. 449–471, Aug. 1999.

[22] J. Dallon, J. Sherratt, P. Maini, and M. Ferguson, "Biological implications of a discrete mathematical model for collagen deposition and alignment in dermal wound repair," *IMA J. Math. Appl. Med. Biol.*, vol. 17, pp. 379–393, Dec. 2000.

[23] A. C. Salazar Coariti, M. S. Fabien, J. Guzman, J. A. McGuire, R. De Vita, and K. C. Toussaint, "Fluid mechanics approach to analyzing collagen fiber organization," *J. Biomed. Opt.*, vol. 27, Jan. 2022.

[24] A. C. Salazar Coariti, M. Fabien, J. Guzmán, and K. Toussaint, "Quantitative

second-harmonic generation imaging analysis based on fluid-dynamics measures," *Frontiers in Optics*, Sept. 2020.

[25] W. Lee, A. O. Moghaddam, Z. Lin, B. L. McFarlin, A. J. Wagoner Johnson, and K. C. Toussaint, "Quantitative classification of 3D collagen fiber organization from volumetric images," *IEEE Trans. Med. Imaging*, vol. 39, pp. 4425–4435, Dec. 2020.

[26] "Introduction to algorithms." https://mitpress.mit.edu/9780262533058/introduction-to-algorithms/, Dec. 2021. Accessed: 2025-7-14.

[27] C. R. Doering and J. D. Gibbon, *Cambridge texts in applied mathematics: Applied analysis of the Navier-stokes equations series number 12*. Cambridge, England: Cambridge University Press, Feb. 2011.

[28] M. Matyka and Z. Koza, "How to calculate tortuosity easily?," AIP, 2012.

CHAPTER 4

Conclusion and Future Work

4.1 Conclusion

This thesis presented a framework for quantifying two and three-dimensional collagen fiber organization using Fourier Transform Second Harmonic Generation (FT-SHG) imaging in conjunction with the Fluid-Inspired Fiber Analysis (FIFA) for quantifying spatial changes in fiber architecture. Our analysis demonstrates that collagen microarchitecture is not only structurally diverse but also encodes meaningful information about tissue state. The development of FIFA provides a new lens through which to assess fiber alignment, directionality, and deformation through spatially resolved metrics. These metrics energy(E), enstrophy (ℓU)) and tortuosity (τ offer quantitative metrics of spatial changes in collage fiber organization and insights on waviness of the fibers.

4.2 Future Work

While this method demonstrated strong potential for spatial fiber organization changes and curvature in native and engineered tissues, several avenues remain for further development and application.

4.2.1 Applications

Future studies should explore the application of the tools developed in this thesis to a range of pathological conditions where collagen type I organization plays a critical role. Diseases such as fibrosis, cancer, osteogenesis imperfecta, and tendinopathy offer opportunities to evaluate the sensitivity and specificity of the fluid-inspired metrics (E, U, and τ) in detecting spatial alterations in the extracellular matrix. It would also be valuable to assess how these metrics respond to biological variables such as sex and disease stage, which could further demonstrate their translational relevance. In addition, in vitro models can be leveraged to mimic in vivo Col-I remodeling under controlled conditions, allowing a more mechanistic understanding of disease progression. These studies could help bridge the gap between collagen structural features and clinically meaningful phenotypes. Ultimately, this framework could support the development of diagnostic tools by identifying pathological states based on quantifiable changes in collagen fiber architecture. Integrating this method with intravital SHG microscopy or cleared whole-organ imaging could extend its biological relevance. Second, FIFA could be expanded to include time-resolved analyses. By analyzing spatial changes in fiber orientation and network topology over time, the method could reveal dynamic matrix reorganization in response to biochemical or mechanical stimuli. This would provide insight into temporal aspects of extracellular matrix remodeling not captured in static analyses

4.2.2 Tool Development

Our goal for the continued development of FIFA is to support and advance research in collagen and other fibrous structures. To facilitate broader adoption, we aim to create a centralized repository with clear documentation to help researchers worldwide access and use the tools. Increasing accessibility could also be achieved by developing a user-friendly graphical interface that enables intuitive interaction with the code. Additionally,

a FIJI/ImageJ plugin could be created to integrate FIFA into widely-used image analysis platforms. From a methodological perspective, streamline generation may benefit from the integration of deep learning to automatically identify the origins and endpoints of fibers. This could enhance our ability to reconstruct collagen microarchitecture with greater fidelity and efficiency. Over time, training such models may also provide new insights into fiber development and dynamics. Additionally, exploring additional fluid-inspired metrics could offer further means to quantify and characterize fiber organization.

4.2.3 Simulations

Our tools have the potential to enhance data-driven computational models by providing high-resolution structural input to refine simulations. Conversely, simulation outputs could be used within our framework to estimate additional matrix properties, such as mechanical behavior. Given the strong link between collagen architecture and mechanical properties, incorporating structural data into simulations could enable more accurate prediction of tissue mechanics. Furthermore, our tools could support the development of predictive models of disease progression or tissue remodeling in response to biochemical stimuli.

Finally, the broader applicability of this framework could be applied to other fibrous ECM proteins, such as elastin or fibrin. Combining FIFA with multimodal imaging approaches (e.g., two-photon fluorescence, third-harmonic generation) could provide a more comprehensive picture of matrix organization and its role in tissue function. In summary, continued refinement and extension of FIFA, both technically and biologically, hold promise for advancing our understanding of tissue structure function relationships in health and disease.

Appendix

Source code

The following appendix contains the source code developed for this work. The code implements the algorithms described in Chapter 2 and 3, including fiber orientation calculation, correction algorithms, fluid inspired metrics and visualization. It is provided here to support reproducibility and allow future extensions.

```
1
2  % Variables
3  % I = Image Path;
4  % dark_th = [0-1](0.2);
5  % iso_th = [0-1](0.3);
6  % xgrid = horizontal grid size (32);
7  % ygrid = vertical grid size (32);
8  function[theta0_all,im,xgrid,ygrid] = Direction(I,dark_th,iso_th,xgrid,
       ygrid)
9
10    im = imread(I)
11
12    figure
13    imshow(im)
14
15     DD = size(im);
16     if numel(DD) == 3
```

```
        im = im(:,:,2);
    end
    im = mat2gray(im);
    [x y] = size(im);

    sigma = 1;
     if sigma*6 <= 3
            fsize = [3 3];
        else
            fsize = ceil([sigma*6, sigma*6]);
        end
    f = fspecial('gaussian',fsize, sigma);
    imfil = imfilter(im,f,'symmetric');

    imt = nan([x y 3]);
    imt(:,:,2) = imfil;

    threshold = dark_th*max(imfil(:));

    D(1) = x - mod(x,xgrid);
    D(2) = y - mod(y,ygrid);

    index = 0;
    num_regions= floor(D(1)/xgrid) * floor(D(2)/ygrid);
    theta0_all=nan(num_regions,1);
    ng_all=nan(num_regions,1);

    for bx = 1:xgrid:D(1)
        for by = 1:ygrid:D(2)
            imtemp = imfil(bx:bx+xgrid-1,by:by+ygrid-1);
            index = index + 1;

```

```
49          if mean(imtemp(:)) >= threshold && mean(imtemp(:) >
    threshold) >= 0.5
50
51              [dx,dy] = gradient(imtemp);
52              temp = ones(xgrid,ygrid);
53              temp = temp - padarray(ones(xgrid-2,ygrid-2),[1 1]);
54              dx = dx.*(1-temp);
55              dy = dy.*(1-temp);
56              dx = dx(imtemp>threshold);
57              dy = dy(imtemp>threshold);
58
59              exclude_points = (dx == 0) & (dy == 0);
60
61              dx(exclude_points) = [];
62              dy(exclude_points) = [];
63
64              if ~isempty(dx)
65
66                  theta = atan2(dx,dy);
67                  theta(theta<0) = theta(theta<0)+pi;
68                  theta = sort(theta);
69
70                  steps = 90;
71                  theta_temp = theta*ones(1,steps);
72                  theta_shift = ones(length(theta),1)*linspace(0,pi-
    pi/steps,steps);
73                  theta_temp = theta_temp+(theta_temp<theta_shift)*pi
    ;
74
75                  theta_mean = angle(sum(exp(1i*theta_temp)));
76                  r = abs(mean(exp(1i*theta_temp)));
77                  circ_var = 1 - r;
78
```

```
                    theta_mean = theta_mean(circ_var == min(circ_var));
                    theta_mean = theta_mean(1);
                    circ_var = min(circ_var);

                    theta_mean(theta_mean < 0) = theta_mean(theta_mean
  < 0) + pi;
                    theta_mean(theta_mean > pi) = theta_mean(theta_mean
   > pi) - pi;


                    if (circ_var > iso_th);
                        %label region as iso
                        ng_all(index) = 5;
                    else
                        %label region as aniso
                        ng_all(index) = 1;
                        theta0_all(index)=theta_mean;
                    end
                else
                    ng_all(index) = 5;
                end
            else
                %label region as dark
                ng_all(index) = -1;
            end


        end
    end
    f1=figure;
    imshow(imt); title('Results');

    iii=0;
```

```
110     dark=0;
111     anisotropic_regions=0;
112     isotropic_regions=0;
113     anisotropy_orientations=[];
114     anis_indices=[];
115     QuivX=[];
116     QuivY=[];
117     Overlay=nan(size(im));
118     OverlayAlphaData=nan(size(im));
119     for cx=1:xgrid:D(1)
120         for cy=1:ygrid:D(2)
121             iii=iii+1;
122             if ng_all(iii)==-1
123                 rectangle('Position',[cy-0.5,cx-0.5,ygrid,xgrid], '
    EdgeColor','yellow','LineStyle','-', 'LineWidth', 1);
124                 Overlay(cx:cx+xgrid-1,cy:cy+ygrid-1)=2;
125                 OverlayAlphaData(cx:cx+xgrid-1,cy:cy+ygrid-1)=0.45; %
    Transparency: 0=transparent; 1=opaque
126                 dark = dark+1;
127             elseif ng_all(iii)>=1 && ng_all(iii)<=4
128                 anisotropy_orientations=[anisotropy_orientations;
    theta0_all(iii)];
129                 anis_indices=[anis_indices iii];
130                 QuivX=[QuivX; cy+ceil(ygrid/2)-ygrid/4*cos(theta0_all(
    iii))];
131                 QuivY=[QuivY; cx+ceil(xgrid/2)+xgrid/4*sin(theta0_all(
    iii))];
132                 rectangle('Position',[cy-0.5,cx-0.5,ygrid,xgrid], '
    EdgeColor','yellow', 'LineStyle','-','LineWidth', 1);
133                 anisotropic_regions = anisotropic_regions+1;
134             else
135                 rectangle('Position',[cy-0.5,cx-0.5,ygrid,xgrid], '
    EdgeColor','yellow', 'LineStyle','-','LineWidth', 1);
```

```
                Overlay(cx:cx+xgrid-1,cy:cy+ygrid-1)=1;
                OverlayAlphaData(cx:cx+xgrid-1,cy:cy+ygrid-1)=0.35;
                isotropic_regions = isotropic_regions+1;
            end
        end
    end

    QuivU=cos(theta0_all);
    QuivV=sin(theta0_all);
    assignin('base','theta0_all',theta0_all);
    hold on;
    quiver(QuivX,QuivY,QuivU(anis_indices),-QuivV(anis_indices)...
        ,0.2,'Color','white','LineWidth',1.5,'MaxHeadSize',0.0);
    hold off;


    num=min(floor(D(1)/xgrid),floor(D(2)/ygrid));
    O=vec2mat(theta0_all,num);



    %Overlay for isotropic and dark regions
    hold on;
    image(Overlay,'AlphaData',OverlayAlphaData);
    colormap([1 0 1;0 1 1]);
    hold off;


    anisotropy_orientations = mod(anisotropy_orientations,pi);
    num_bars= 360/5;
    steps = 90;
    theta_temp = anisotropy_orientations*ones(1,steps);
```

```
168     theta_shift = ones(length(anisotropy_orientations),1)*linspace(0,pi
    -pi/steps,steps);
169     theta_temp = theta_temp+(theta_temp<theta_shift)*pi;
170
171     theta_mean = angle(sum(exp(1i*theta_temp)));
172     r = abs(mean(exp(1i*theta_temp)));
173     circ_var = 1 - r;
174
175     theta_mean = theta_mean(circ_var == min(circ_var));
176     theta_mean = theta_mean(1);
177     circ_var = min(circ_var);
178
179     theta_mean(theta_mean < 0) = theta_mean(theta_mean < 0) + pi;
180     theta_mean(theta_mean > pi) = theta_mean(theta_mean > pi) - pi;
181
182     theta_mean = theta_mean/pi*180;
183     f2=figure;
184     rose([anisotropy_orientations;anisotropy_orientations+pi], num_bars
    *2);
185     yl = get(gca,'Ylim');
186     text(yl(yl>0)/2*1.1,yl(yl>0)/2*1.7,sprintf('Average Orientation =
    %.1f', theta_mean),'FontSize',11, 'Color', 'blue');
187     text(yl(yl>0)/2*1.1,yl(yl>0)/2*1.5,sprintf('Circular variance = %.3
    f', circ_var*2),'FontSize',11, 'Color', 'blue');
188     title('Polar Histogram of orientations from anisotropic regions');
189
190     f3=figure;
191     br=bar([dark anisotropic_regions isotropic_regions],0.35);
192     set(gca,'XTickLabel',{'Dark';'Anisotropic';'Isotropic'},'FontSize'
    ,8)
193     yl=get(gca, 'Ylim');
194     ylim([0, yl(2)*1.2]);
195
```

```
196
197     set(br,'FaceColor',[0 1 1]);
198
199
200     text(1,dark,num2str(dark),'FontSize',9,'Color','blue',...
201         'FontWeight','bold','HorizontalAlignment','center','
    VerticalAlignment','bottom');
202     text(2,anisotropic_regions,num2str(anisotropic_regions),'FontSize'
    ,9,'Color','blue',...
203         'FontWeight','bold','HorizontalAlignment','center','
    VerticalAlignment','bottom');
204     text(3,isotropic_regions,num2str(isotropic_regions),'FontSize',9,'
    Color','blue',...
205         'FontWeight','bold','HorizontalAlignment','center','
    VerticalAlignment','bottom');
206     set(gcf,'position',[500 500 300 200]);  %setting to a certain size
207 end
```

Listing 4.1: Calculation of 2D Directions

```
1 %Variables
2 %angleInDegrees = Angle from Direction function
3 %sideSize_x = 32
4 %sideSize_y = 32
5 %gridS = 32(same as above if square grid)
6 %rn = 1
7 function[FD_KEnergy,FD_Enstrophy,Xdata,Ydata]=FD_v1(angleInDegrees,
    sideSize_x,sideSize_y,gridS,rn,im)
8
9 h1 = 1/sideSize_x;
10 h2 = 1/sideSize_y;
11 angle = repelem(angleInDegrees,rn,rn);
12 f1 = (cosd(angle)); % x component of vector field
13 f2 = (sind(angle)); % y component of vector field
```

```
14
15
16 %FINITE DIFFERENCE FOR (d/dx) f (FX)
17 FXf1 = (f1(:,2:end-0)-f1(:,1:end-1))/h1; %Forward Difference (f(a+h,b)-
       f(a,b))/h1
18 FXf2 = (f2(:,2:end-0)-f2(:,1:end-1))/h1; %Forward Difference (f(a,b)-f(
       a-h,b))/h1
19 BXf1 = (f1(:,end)-f1(:,end-1))/h1; % Backward Difference
20 BXf2 = (f2(:,end)-f2(:,end-1))/h1; % Backward Difference
21 F1_x = [FXf1 BXf1];
22 F2_x = [FXf2 BXf2];
23
24 %FINITE DIFFERENCE FOR (d/dy) f (FY)
25 FYf1 = (f1(2:end-0,:)-f1(1:end-1,:))/h2; %Forward Difference (f(a,b+h)-
       f(a,b))/h2
26 FYf2 = (f2(2:end-0,:)-f2(1:end-1,:))/h2; %Forward Difference (f(a,b)-f(
       a,b-h))/h2
27 BYf1 = (f1(end,:)-f1(end-1,:))/h2; % Backward Difference
28 BYf2 = (f2(end,:)-f2(end-1,:))/h2; % Backward Difference
29 F1_y = [FYf1; BYf1];
30 F2_y = [FYf2; BYf2];
31 FD_curl = F2_x - F1_y;
32
33 %Energy
34 F12xy2 = F1_x.^2 + F2_x.^2 + F1_y.^2 + F2_y.^2;
35 FD_KEnergy = 0.5*h1*h2*sum(F12xy2(:),'omitnan');
36
37 %Enstrophy
38 FD_curl2 = FD_curl.^2;
39 FD_Enstrophy = h1*h2*sum(FD_curl2(:),'omitnan');
40
41 u=f1;v=f2;
42 dimx = size(angle,1);
```

```
dimy = size(angle,2);
figure
sizeby2 = gridS/rn

%Streamlines

x = 1:sizeby2:dimx*sizeby2;
y = 1:sizeby2:dimy*sizeby2;
[X,Y] = meshgrid(x,y);

h11 = streamline(stream2(X,Y,u,v,X,Y))

xlabel('x (\mum)')
ylabel('y (\mum)')
    plt2 =gca;
    set(plt2, 'FontName', 'Arial')
    set(plt2,'fontweight','bold','fontsize',10)
    set(plt2, 'xtick', 0:100:sideSize_x*sizeby2);
    set(plt2, 'ytick', 0:100:sideSize_y*sizeby2);
   set(h11,'Color','m')


    Xdata = get(h11,'Xdata');
    Ydata = get(h11,'Ydata');

    for ind = 1:length(Xdata)
        if size(Xdata{ind,1},2)==1
            Xdata{ind,1} = NaN;
            Ydata{ind,1} = NaN;
        end
    end

```

```
end
```

Listing 4.2: Fluid-Inspired Quantification of 2D Fiber Organization

```
%Main
I = 'C:\Users\asalaz12\Downloads\2D_FIFA\SHG_1.tif';
im = imread(I);
dark_th = 0.2;
iso_th = 0.3;
xgrid = 32;
ygrid = 32;


Direction(I,dark_th,iso_th,xgrid,ygrid);


%%
%sideSize_x = xgrid;
%sideSize_y = ygrid;
ang_deg = rad2deg(theta0_all);

sideSize_x = fix(size(im,2)/xgrid);
sideSize_y = fix(size(im,1)/ygrid);

c=transpose(reshape(ang_deg,[sideSize_x,sideSize_y]));
[x, y] = meshgrid(1:sideSize_x,1:sideSize_y);
b = flip(c);

angleInDegrees = b;

gridS = 32;
rn = 1;

[FD_KEnergy,FD_Enstrophy,Xdata,Ydata]=FD_v1(angleInDegrees,sideSize_x,
```

```
    sideSize_y,gridS,rn,im);
```

Listing 4.3: Main Environment for 2D Analysis

```
function [vec,dark,iso,not_ana,f3D_avail] = fiber3D_revised()

clear all;
defaultvalues;

while 1

    %% Menu
    not_avail = [];
    if isempty(vec)
        not_avail = [not_avail, 12, 14];
    end
    if isempty(dark) || isempty(iso) || isempty(not_ana)
        not_avail = [not_avail, 12, 13, 15];
    end
    if isempty(im)
        not_avail = [not_avail, 0];
    end
    if isempty(im2)
        not_avail = [not_avail, 11];
    end

    clc;
    fprintf('Options:\n');
    fprintf('1. Enter out-of-plane orientation threshold [current: %0.3
    f rad]\n',asin(color_th));
    fprintf('2. Enter filter size (current: %i pixels; integer only)\n'
    ,filsize);
    fprintf('3. Enter Sigma for filter (current: %0.3f pixels)\n',sig);
    fprintf('4. Enter upper threshold (current: %0.3f)\n',th_up);
```

```
29      fprintf('5. Enter lower threshold (current: %0.3f)\n',th_low);
30      fprintf('6. Enter grid size (current: %i pixels; integer only)\n',
    gridsize);
31      fprintf('7. Enter dark threshold (current: %0.3f)\n',dark_th);
32      fprintf('8. Enter isotropic threshold (current: %0.3f)\n',iso_th);
33      fprintf('9. Enter number of iteration (1-4)(current: %i; integer
    only)\n',nint);
34      fprintf('10. Enter orientation filter size (current: %0.1f pixels;
    multiples of 0.5 only)\n',dsize);
35      fprintf('11. Show Canny Surface? (Y/N) [current: %s;]\n [Out-of-
    plane orientation threshold: %0.3f rad]\n',surfplot,asin(color_th));
36      fprintf('12. Show quiver plot? (Y/N) [current: %s]\n',qplot);
37      fprintf('13. Show overall analyzed plot? (Y/N) [current: %s]\n',
    anaplot);
38      fprintf('14. Show histogram of 3D orientation? (Y/N) [current: %s]\
    n',hplot);
39      fprintf('15. Show histogram of number of threshold regions? (Y/N) [
    current: %s]\n',thplot);
40      fprintf('16. Reload image stack.\n');
41      fprintf('17. Load previous data.\n');
42      fprintf('18. Load previous setting.\n');
43      fprintf('19. Close all previous figures\n');
44      fprintf('20. Load default.\n');
45      fprintf('21. Save data and setting.\n');
46      fprintf('22. Command mode. (For modifying plots with general MATLAB
        commands)\n');
47      if ~isempty(not_avail)
48          fprintf('The following options are not available:');
49          disp(sort(unique(not_avail)));
50          fprintf('\n');
51      end
52      fprintf('Enter number to change settings,');
53      if ~isempty(im)
```

```
54          fprintf(' enter ''0'' to analyze loaded image stack,');
55      else
56          not_avail = [not_avail, 0];
57      end
58      option = input(' or enter ''q'' to exit: ','s');
59      opt = str2num(option);
60      if ~isempty(opt) && sum(not_avail==opt) == 0
61
62          switch opt
63              case 0
64
65                  r =(dsize-1)/2;
66                  type = 2;
67                  tic;
68
69                  %   Initialize all variables
70                  [w h d] = size(im);
71                  icount = ceil(w/gridsize);
72                  jcount = ceil(h/gridsize);
73                  kcount = ceil(d/gridsize);
74                  total = icount*jcount*kcount;
75                  vec = zeros(icount,jcount,kcount,3);
76                  vec1 = zeros(icount,jcount,kcount,3);
77                  vec2 = zeros(icount,jcount,kcount,3);
78                  dark = zeros(icount,jcount,kcount);
79                  iso = zeros(icount,jcount,kcount);
80                  iso2 = zeros(icount,jcount,kcount);
81                  ang = 1000*ones(icount,jcount,kcount,2);
82                  ang1 = 1000*ones(icount,jcount,kcount,2);
83                  ang2 = 1000*ones(icount,jcount,kcount,2);
84                  not_ana = zeros(icount,jcount,kcount);
85
86                  %   Perform calculation
```

```
87                  im2 = ~true(w,h,d);
88                  for i = 1:icount
89                      for j = 1:jcount
90                          for k = 1:kcount
91                              b = toc;
92                              x1 = 3; y1 = 3; z1 = 3;
93                              x2 = gridsize+2; y2 = gridsize+2; z2 =
    gridsize+2;
94                              irange = ((i-1)*gridsize+1):(i*gridsize);
95                              jrange = ((j-1)*gridsize+1):(j*gridsize);
96                              krange = ((k-1)*gridsize+1):(k*gridsize);
97                              tirange = ((i-1)*gridsize-1):(i*gridsize+2)
    ;
98                              tjrange = ((j-1)*gridsize-1):(j*gridsize+2)
    ;
99                              tkrange = ((k-1)*gridsize-1):(k*gridsize+2)
    ;
100                             %   Begining
101                             if i == 1
102                                 tirange = ((i-1)*gridsize+1):(i*
    gridsize+2);
103                                 x1 = 1; x2 = gridsize;
104                             end
105                             if j == 1
106                                 tjrange = ((j-1)*gridsize+1):(j*
    gridsize+2);
107                                 y1 = 1; y2 = gridsize;
108                             end
109                             if k == 1
110                                 tkrange = ((k-1)*gridsize+1):(k*
    gridsize+2);
111                                 z1 = 1; z2 = gridsize;
112                             end
```

```
113                             % End
114                             if i >= w/gridsize
115                                 irange = ((i-1)*gridsize+1):w;
116                                 tirange = ((i-1)*gridsize-1):w;
117                                 x2 = length(tirange);
118                             end
119                             if j >= h/gridsize
120                                 jrange = ((j-1)*gridsize+1):h;
121                                 tjrange = ((j-1)*gridsize-1):h;
122                                 y2 = length(tjrange);
123                             end
124                             if k >= d/gridsize
125                                 krange = ((k-1)*gridsize+1):d;
126                                 tkrange = ((k-1)*gridsize-1):d;
127                                 z2 = length(tkrange);
128                             end
129                             % If there is only one cell to calculate
130                             if icount == 1
131                                 irange = 1:w;
132                                 tirange = 1:w;
133                                 x1 = 1; x2 = w;
134                             end
135                             if jcount == 1
136                                 jrange = 1:h;
137                                 tjrange = 1:h;
138                                 y1 = 1; y2 = h;
139                             end
140                             if kcount == 1
141                                 krange = 1:d;
142                                 tkrange = 1:d;
143                                 z1 = 1; z2 = d;
144                             end
145                             % If tirange is out of range, seems to be
```

```
146                             % unnecessary
147                             if tirange(end) > w
148                                 tirange(end) = [];
149                             end
150                             if tjrange(end) > h
151                                 tjrange(end) = [];
152                             end
153                             if tkrange(end) > d
154                                 tkrange(end) = [];
155                             end
156                             imtemp = im(tirange,tjrange,tkrange);
157                             % Dark pixels
158                             if min(size(imtemp)) < min_grid
159                                 not_ana(i,j,k) = 1;
160                             else
161                                 if mean2(imtemp) > dark_th
162                                     imtemp = canny3D(imtemp, filsize,
    0.5, th_up, th_low);
163                                     im2(irange,jrange,krange) = imtemp(
    x1:x2,y1:y2,z1:z2);
164                                     [vec(i,j,k,:),vec1(i,j,k,:),vec2(i,
    j,k,:),ang(i,j,k,:),ang1(i,j,k,:),ang2(i,j,k,:),iso(i,j,k),iso2(i,j,
    k)] = findorientation2(im2(irange,jrange,krange),sig,r,type,nint,
    iso_th);
165                                 else
166                                     dark(i,j,k) = 1;
167                                 end
168                             end
169                             completion = (((i-1)*jcount*kcount + ((j-1)
    *kcount+k))/total*100);
170                             clc;
171                             fprintf('%.2f%% completed......\n',
    completion);
```

```
172                         end
173                     end
174                 end
175                 toc;
176 
177                 %% Save results
178                 assignin('base','im',im);
179                 assignin('base','canny',im2);
180                 surfsave = input('Save 3D Canny Surface images? (Y/N) [
    default: N]','s');
181                 if isempty(surfsave)
182                     surfsave = 'N';
183                 end
184 
185                 if surfsave ~= 'N'
186                     savestack(im2);
187                 end
188 
189                 vecsave = input('Save 3D orientation results? (Y/N) [
    default: Y]','s');
190                 if isempty(vecsave)
191                     vecsave = 'Y';
192                 end
193 
194                 if vecsave == 'Y'
195                     uisave({'vec','vec1','vec2','dark','ang','ang1','
    ang2','iso','iso2','not_ana','w','h','d','gridsize','total'},'data')
    ;
196                 end
197 
198             case 1
199 
200                 color_th = sin(input('1. Enter an out-of-plane
```

```
orientation threshold (0 - pi/2 rad):'));
201                 color_th=2;
202                 if isempty(rselect)
203                     color_th = sin(90/180*pi);
204                     color_th=2;
205                 end
206
207             case 2
208
209                 filsize = input('2. Enter filter size (default: 9
pixels; integer only):');
210                 if isempty(filsize)
211                     filsize = 9;
212                 end
213
214             case 3
215
216                 sig = input('3. Enter Sigma for filter (default: 1.5
pixels):');
217                 if isempty(sig)
218                     sig = 1.5;
219                 end
220
221             case 4
222
223                 th_up = input('4. Enter upper threshold (default: 0.02)
:');
224                 if isempty(th_up)
225                     th_up = 0.02;
226                 end
227                 % update th_low
228                 th_low = th_up*0.5;
229
```

```
230             case 5
231 
232                 th_low = input('5. Enter lower threshold (default: 50%
    of upper threshold):');
233                 if isempty(th_low)
234                     th_low = th_up*0.5;
235                 end
236 
237             case 6
238 
239                 gridsize = input('6. Enter grid size (default: 44
    pixels; min: 8 pixels; max: 128 pixels; integer only):');
240                 if isempty(gridsize)
241                     gridsize = 44;
242                 end
243                 if gridsize < min_grid
244                     gridsize = min_grid;
245                 end
246                 if gridsize > max_grid
247                     gridsize = max_grid;
248                 end
249                 % update dsize
250                 dsize = gridsize/2;
251 
252             case 7
253 
254                 dark_th = input('7. Enter dark threshold (default: 0):'
    );
255                 if isempty(dark_th)
256                     dark_th = 0;
257                 end
258 
259             case 8
```

```
260
261             iso_th = input('8. Enter isotropic threshold (default:
    1):');
262             if isempty(iso_th)
263                 iso_th = 1;
264             end
265
266         case 9
267
268             nint = input('9. Enter number of iteration (1-4)(
    default: 2; integer only):');
269             if isempty(nint)
270                 nint = 2;
271             end
272
273         case 10
274
275             dsize = input('10. Enter orientation filter size (
    default: half of grid size; multiples of 0.5 only):');
276             if isempty(dsize)
277                 dsize = gridsize/2;
278             end
279
280         case 11
281
282             surfplot = input('11. Show Canny Surface? (Y/N) [
    default: N]','s');
283             if isempty(surfplot)
284                 surfplot = 'N';
285             end
286
287             %% surfplot
288             if surfplot ~= 'N'
```

```
289                     implay(im2);
290                 end
291
292                 surfplot = 'N';
293
294             case 12
295
296                 qplot = input('12. Show quiver plot? (Y/N) [default: N]
    ','s');
297                 if isempty(qplot)
298                     qplot = 'N';
299                 end
300
301                 %% 3D quiver plot
302                 % uncomment the *lines to generate movies for the plots
303                 if qplot ~= 'N'
304
305 %*                      clear mov;
306                     [icount,jcount,kcount,~] = size(vec);
307                     w = icount*gridsize;
308                     h = jcount*gridsize;
309                     d = kcount*gridsize;
310 %*                      mov = avifile('qplot.avi','compression','None','
    fps' ,1);
311 %*                      figure('Renderer','zbuffer');
312                     hold on;
313                     subplot(2,3,[1 2 4 5])
314                     axis([0 h 0 w 0 d]);
315 %                     daspect([1 1 1]);
316 %*                      set(gcf,'position',[400 400 720 480]);
317 %*                      set(gca,'XTickLabel','');
318 %*                      set(gca,'YTickLabel','');
319 %*                      set(gca,'ZTickLabel','');
```

```
%*                        set(gca,'NextPlot','replaceChildren');
                       subplot(2,3,3)
                       axis([0 h 0 w 0 d]);
%                        daspect([1 1 1]);
%*                        set(gcf,'position',[400 400 720 480]);
%*                        set(gca,'XTickLabel','');
%*                        set(gca,'YTickLabel','');
%*                        set(gca,'ZTickLabel','');
%*                        set(gca,'NextPlot','replaceChildren');

                       % Generate temporary variables for quiver plot
                       % Orientation data is obtained from analysis or
    data
                       % file
                       vectemp = zeros(3,1);
                       vectemp1 = zeros(3,1);
                       vectemp2 = zeros(3,1);
                       uq = [];
                       vq = [];
                       wq = [];
                       iq = [];
                       jq = [];
                       kq = [];
                       uq11 = [];
                       vq11 = [];
                       wq11 = [];
                       iq11 = [];
                       jq11 = [];
                       kq11 = [];
                       uq22 = [];
                       vq22 = [];
                       wq22 = [];
                       iq22 = [];
```

```
352                     jq22 = [];
353                     kq22 = [];
354                     uq2 = [];
355                     vq2 = [];
356                     wq2 = [];
357                     iq2 = [];
358                     jq2 = [];
359                     kq2 = [];
360                     ind2 = 0;
361                     ind22 = 0;
362 
363                     for k = 1:kcount
364                         ind1 = ind2+1;
365                         ind12 = ind22+1;
366                         for j = 1:jcount
367                             for i = 1:icount
368                                 if dark(i,j,k) == 0 && iso(i,j,k) == 0
369                                     vectemp(:) = vec(i,j,k,:);
370                                     vectemp1(:) = vec1(i,j,k,:);
371                                     vectemp2(:) = vec2(i,j,k,:);
372                                     if norm(vectemp) > 0 || norm(
    vectemp1) > 0 || norm(vectemp2) > 0
373 
374 
375                                         if abs(vec(i,j,k,3)) < 1.5 ||
    abs(vec1(i,j,k,3)) < 1.5 || abs(vec2(i,j,k,3)) < 1.5
376                                             ind2 = ind2+1;
377                                             uq(end+1) = vec(i,j,k,1)/
    norm(vectemp)*gridsize;
378                                             vq(end+1) = vec(i,j,k,2)/
    norm(vectemp)*gridsize;
379                                             wq(end+1) = vec(i,j,k,3)/
    norm(vectemp)*gridsize;
```

```
380                                         uq11(end+1) = vec1(i,j,k,1)
    /norm(vectemp1)*gridsize;
381                                         vq11(end+1) = vec1(i,j,k,2)
    /norm(vectemp1)*gridsize;
382                                         wq11(end+1) = vec1(i,j,k,3)
    /norm(vectemp1)*gridsize;
383                                         uq22(end+1) = vec2(i,j,k,1)
    /norm(vectemp2)*gridsize;
384                                         vq22(end+1) = vec2(i,j,k,2)
    /norm(vectemp2)*gridsize;
385                                         wq22(end+1) = vec2(i,j,k,3)
    /norm(vectemp2)*gridsize;
386                                         iq(end+1) = (i-1/2)*
    gridsize-uq(end)/2;
387                                         jq(end+1) = (j-1/2)*
    gridsize-vq(end)/2;
388                                         kq(end+1) = (k-1/2)*
    gridsize-wq(end)/2;
389                                         iq11(end+1) = (i-1/2)*
    gridsize-uq11(end)/2;
390                                         jq11(end+1) = (j-1/2)*
    gridsize-vq11(end)/2;
391                                         kq11(end+1) = (k-1/2)*
    gridsize-wq11(end)/2;
392                                         iq22(end+1) = (i-1/2)*
    gridsize-uq22(end)/2;
393                                         jq22(end+1) = (j-1/2)*
    gridsize-vq22(end)/2;
394                                         kq22(end+1) = (k-1/2)*
    gridsize-wq22(end)/2;
395                                     else
396                                         ind22 = ind22+1;
397                                         uq2(end+1) = vec(i,j,k,1)/
```

```
norm(vectemp)*gridsize;
398                                                 vq2(end+1) = vec(i,j,k,2)/
norm(vectemp)*gridsize;
399                                                 wq2(end+1) = vec(i,j,k,3)/
norm(vectemp)*gridsize;
400                                                 iq2(end+1) = (j-1/2)*
gridsize-uq2(end)/2;
401                                                 jq2(end+1) = (i-1/2)*
gridsize-vq2(end)/2;
402                                                 kq2(end+1) = (k-1/2)*
gridsize-wq2(end)/2;
403                                             end
404                                         end
405                                     end
406                                 end
407                             end
408                             subplot(2,3,[1 2 4 5]),quiver3(iq,jq,kq,uq,vq,
wq,'Color',[1,0,0],'linewidth',2,'AutoScale','off');
409                             hold on,quiver3(iq2,jq2,kq2,uq2,vq2,wq2,'Color'
,[0,1,0],'linewidth',2,'AutoScale','off'), daspect([1 1 1]);
410                             hold on,quiver3(iq11,jq11,kq11,uq11,vq11,wq11,'
Color',[0,1,0],'linewidth',2,'AutoScale','off');
411                             hold on,quiver3(iq22,jq22,kq22,uq22,vq22,wq22,'
Color',[0,0,1],'linewidth',2,'AutoScale','off');
412                             view(-35, 35);
413
414                             subplot(2,3,3),quiver3(iq(ind1:ind2),jq(ind1:
ind2),kq(ind1:ind2),uq(ind1:ind2),vq(ind1:ind2),wq(ind1:ind2),'Color
',[1,0,0],'linewidth',2,'AutoScale','off');
415                             hold on,quiver3(iq2(ind12:ind22),jq2(ind12:
ind22),kq2(ind12:ind22),uq2(ind12:ind22),vq2(ind12:ind22),wq2(ind12:
ind22),'Color',[0,1,0],'linewidth',2,'AutoScale','off');
416                             hold on,quiver3(iq11(ind1:ind2),jq11(ind1:ind2)
```

```
    ,kq11(ind1:ind2),uq11(ind1:ind2),vq11(ind1:ind2),wq11(ind1:ind2),'
    Color',[0,1,0],'linewidth',2,'AutoScale','off');
417                         hold on,quiver3(iq22(ind1:ind2),jq22(ind1:ind2)
    ,kq22(ind1:ind2),uq22(ind1:ind2),vq22(ind1:ind2),wq22(ind1:ind2),'
    Color',[0,0,1],'linewidth',2,'AutoScale','off');
418                         view(90,90);daspect([1 1 1]);
419
420                         if ~isempty(im) && k ~= kcount
421                             subplot(2,3,6),imshow(im(:,:,(k-1)*gridsize
    +gridsize/2));
422                         end
423                         if ~isempty(im) && k == kcount
424                             [~,~,dd] = size(im);
425                             subplot(2,3,6),imshow(im(:,:,floor((dd+(k
    -1)*gridsize)/2)));
426                         end
427 %*                          frame = getframe(gcf);
428 %*                          mov = addframe(mov,frame);
429                     end
430 %*                      mov = close(mov);
431 %*                     fprintf('The figure is saved as qplot.avi in the
    application directory');
432                 end
433
434                 qplot = 'N';
435
436             case 13
437
438                 anaplot = input('13. Show overall analyzed plot? (Y/N)
    [default: N]','s');
439                 if isempty(anaplot)
440                     anaplot = 'N';
441                 end
```

```
442
443                 %% plot of labeled regions. The plotcude function is
    not optimzed for speed. This option
444                 %% may take a long time to execute
445                 % uncomment the *lines to generate movies for the plots
446                 if anaplot ~= 'N'
447                     [icount,jcount,kcount] = size(dark);
448                     D1 = [icount,jcount,kcount];
449                     [icount,jcount,kcount] = size(iso);
450                     D2 = [icount,jcount,kcount];
451                     [icount,jcount,kcount] = size(iso2);
452                     D4 = [icount,jcount,kcount];
453                     [icount,jcount,kcount] = size(not_ana);
454                     D3 = [icount,jcount,kcount];
455
456                     if sum(D1==D2 & D1 == D3) == 3
457
458 %*                      clear mov;
459                       w = icount*gridsize;
460                       h = jcount*gridsize;
461                       d = kcount*gridsize;
462 %*                      mov = avifile('anaplot.avi','compression','
    None','fps',1);
463 %*                      figure('Renderer','zbuffer');
464                       hold on;
465                       subplot(2,3,[1 2 4 5])
466                       view(-35, 35);
467                       axis([0 w 0 h 0 d]);
468                       daspect([1 1 1]);
469 %*                      set(gcf,'position',[400 400 720 480]);
470 %*                      set(gca,'XTickLabel','');
471 %*                      set(gca,'YTickLabel','');
472 %*                      set(gca,'ZTickLabel','');
```

```
473 %*                          set(gca,'NextPlot','replaceChildren');
474                          subplot(2,3,3)
475                          axis([0 w 0 h 0 d]);
476                          daspect([1 1 1]);
477 %*                          set(gcf,'position',[400 400 720 480]);
478 %*                          set(gca,'XTickLabel','');
479 %*                          set(gca,'YTickLabel','');
480 %*                          set(gca,'ZTickLabel','');
481 %*                          set(gca,'NextPlot','replaceChildren');
482
483                          for k = 1:kcount
484                              for j = 1:jcount
485                                  for i = 1:icount
486
487                                      xl = (i-1)*gridsize;
488                                      yl = (j-1)*gridsize;
489                                      zl = (k-1)*gridsize;
490                                      edges = [gridsize gridsize gridsize
    ];
491                                      if i >= w/gridsize
492                                          edges(1) = w-(i-1)*gridsize;
493                                      end
494                                      if j >= h/gridsize
495                                          edges(2) = h-(j-1)*gridsize;
496                                      end
497                                      if k >= d/gridsize
498                                          edges(3) = d-(k-1)*gridsize;
499                                      end
500                                      % if unanalyzed, do not label and
    label
501                                      % the rest
502                                      if iso(i,j,k) ~= 1 && dark(i,j,k)
    ~= 1 && not_ana(i,j,k) ~= 1 && iso2(i,j,k) ~= 1
```

```
503                                         subplot(2,3,[1 2 4 5]),plotcube
    (edges,[xl yl zl],0.5,[0 1 0]);
504                                         subplot(2,3,3),plotcube(edges,[
    xl yl zl],0.5,[0 1 0]);
505                                     end
506                                     if iso(i,j,k) == 1
507                                         subplot(2,3,[1 2 4 5]),plotcube
    (edges,[xl yl zl],0.5,[1 0 0]);
508                                         subplot(2,3,3),plotcube(edges,[
    xl yl zl],0.5,[1 0 0]);
509                                     end
510                                     if dark(i,j,k) == 1
511                                         subplot(2,3,[1 2 4 5]),plotcube
    (edges,[xl yl zl],0.5,[0 0 1]);
512                                         subplot(2,3,3),plotcube(edges,[
    xl yl zl],0.5,[0 0 1]);
513                                     end
514                                     if iso2(i,j,k) == 1
515                                         subplot(2,3,[1 2 4 5]),plotcube
    (edges,[xl yl zl],0.5,[0 1 1]);
516                                         subplot(2,3,3),plotcube(edges,[
    xl yl zl],0.5,[0 1 1]);
517                                     end
518                                     view(-35, 35);
519 %                                      view(90,90);daspect([1 1 1]);
520 %                                    if not_ana(i,j,k) == 1
521 %                                        subplot(2,3,[1 2 4 5]),
    plotcube(edges,[xl yl zl],0.1,[0.25 0.25 0.25]);
522 %                                        subplot(2,3,3),plotcube(edges
    ,[xl yl zl],0.1,[0.25 0.25 0.25]);
523 %                                    end
524                                     subplot(2,3,3),hold on;
525                                 end
```

```
526                             end
527                             subplot(2,3,3),view(90,90);
528                             if ~isempty(im) && k ~= kcount
529                                 subplot(2,3,6),imshow(im(:,:,(k-1)*
    gridsize+gridsize/2));
530                             end
531                             if ~isempty(im) && k == kcount
532                                 [~,~,dd] = size(im);
533                                 subplot(2,3,6),imshow(im(:,:,floor((dd
    +(k-1)*gridsize)/2)));
534                             end
535 %*                              frame = getframe(gcf);
536 %*                              mov = addframe(mov,frame);
537                             subplot(2,3,3),hold off;
538                         end
539 %*                          mov = close(mov);
540 %*                          fprintf('The figure is saved as anaplot.avi
    in the application directory');
541                         else
542                             fprintf('Inconsistent data set for regions plot
    !\n');
543                         end
544                 end
545
546                 anaplot = 'N';
547
548             case 14
549
550                 hplot = input('14. Show histogram of 3D orientation? (Y
    /N) [default: N]','s');
551                 if isempty(hplot)
552                     hplot = 'N';
553                 end
```

```
554
555
556                 if hplot ~= 'N'
557                     zang = asin(vec(:,:,:,3));
558                     zang = zang(sum(vec.^2,4)>0);
559                     % Change to degrees
560                     zang = 90-zang/pi*180;
561                     xyang = atan2(vec(:,:,:,1),vec(:,:,:,2));
562                     xyang = xyang(sum(vec.^2,4)>0);
563                     xyang = (xyang+pi/2)/pi*180;
564                     xyang(xyang<0) = xyang(xyang<0)+180;
565                     xyang(xyang>180) = xyang(xyang>180)-180;
566                     switch nint
567                         case 1
568                             bin = 30;
569                         case 2
570                             bin = 5;
571                         case 3
572                             bin = 1;
573                         case 4
574                             bin = 0.5;
575                     end
576                     figure;
577                     hist(zang,1:bin:180);
578                     yl=get(gca, 'Ylim');
579                     mh = nanmean(zang);
580                     sh = nanstd(zang);
581                     text(20,yl(2)*1.15, sprintf('Mean = %.1f', mh),'
    FontSize',12, 'Color', 'blue');
582                     text(20,yl(2)*1.07, sprintf('Spread = %.1f', sh),'
    FontSize',12, 'Color', 'blue');
583                     %title('\it\phi\rm\bf : orientation with respect to
     the xy-plane','FontSize',13,'FontWeight','bold');
```

```
584                     xlabel('\it\phi\rm\bf (\circ)','FontSize',14,'
    FontWeight','bold');
585                     ylabel('COUNTS','FontSize',12,'FontWeight','bold');
586                     xlim([0 180]);
587                     ylim([0, yl(2)*1.2]);
588                     set(gcf,'position',[400 400 400 300]);
589                     figure;
590                     hist(xyang,1:bin:180);
591                     mh = nanmean(xyang);
592                     sh = nanstd(xyang);
593                     yl=get(gca, 'Ylim');
594                     text(110,yl(2)*1.15, sprintf('Mean = %.1f', mh),'
    FontSize',12, 'Color', 'blue');
595                     text(110,yl(2)*1.07, sprintf('Spread = %.1f', sh),'
    FontSize',12, 'Color', 'blue');
596                     %title('\it\theta\rm\bf : orientation on the xy-
    plane','FontSize',13,'FontWeight','bold');
597                     xlabel('\it\theta\rm\bf (\circ)','FontSize',14,'
    FontWeight','bold');
598                     ylabel('COUNTS','FontSize',12,'FontWeight','bold');
599                     xlim([0 180]);
600                     ylim([0, yl(2)*1.2]);
601                     set(gcf,'position',[400 400 400 300]);
602                 end
603
604                 hplot = 'N';
605
606             case 15
607
608                 thplot = input('15. Show histogram of number of
    threshold regions? (Y/N) [default: N]','s');
609                 if isempty(thplot)
610                     thplot = 'N';
```

```
611                 end
612
613                 %% thplot
614                 if thplot ~= 'N'
615                     [icount,jcount,kcount] = size(dark);
616                     D1 = [icount,jcount,kcount];
617                     [icount,jcount,kcount] = size(iso);
618                     D2 = [icount,jcount,kcount];
619                     [icount,jcount,kcount] = size(not_ana);
620                     D3 = [icount,jcount,kcount];
621                     if sum(D1==D2 & D1 == D3) == 3
622                         figure;
623                         dark_reg = sum(dark(:));
624                         iso_reg = sum(iso(:));
625                         iso2_reg = sum(iso2(:));
626                         na_reg = sum(not_ana(:));
627                         ai_reg = total - dark_reg - iso_reg - na_reg -
    iso2_reg;
628                         th = [ai_reg,iso_reg,iso2_reg,dark_reg];
629                         bar(th,0.35,'g');
630                         yl=get(gca, 'Ylim');
631                         ylim([0, yl(2)*1.2]);
632                         set(gca,'XTickLabel',{'ANISOTROPIC';'ISOTROPIC'
    ;'ISOTROPIC2';'DARK'},'FontSize',11,'FontWeight','bold');
633                         text(1,ai_reg,num2str(ai_reg),'FontSize',12,'
    Color','blue',...
634                         'FontWeight','bold','HorizontalAlignment','
    center','VerticalAlignment','bottom');
635                         text(2,iso_reg,num2str(iso_reg),'FontSize',12,'
    Color','blue',...
636                         'FontWeight','bold','HorizontalAlignment','
    center','VerticalAlignment','bottom');
637                         text(3,iso2_reg,num2str(iso2_reg),'FontSize'
```

```
,12,'Color','blue',...
638                         'FontWeight','bold','HorizontalAlignment','
center','VerticalAlignment','bottom');
639                         text(4,dark_reg,num2str(dark_reg),'FontSize'
,12,'Color','blue',...
640                         'FontWeight','bold','HorizontalAlignment','
center','VerticalAlignment','bottom');
641                     %     text(4,na_reg,num2str(na_reg),'FontSize',9,'
Color','blue',...
642                     %     'FontWeight','bold','HorizontalAlignment','
center','VerticalAlignment','bottom');
643
644                         ylabel('COUNTS','FontSize',12,'FontWeight','
bold');title('LABELED REGIONS','FontSize',14,'FontWeight','bold');
645                         set(gcf,'position',[400 400 400 300]);  %
setting to a certain size
646
647                     else
648                         fprintf('Inconsistent data set for regions
histogram!\n');
649                     end
650                 end
651
652                 thplot = 'N';
653
654             case 16
655
656                 %% Read stack
657                 im = readstack;
658
659             case 17
660
661                 %% Load previous data
```

```
662                 [FileName,PathName,~] = uigetfile('data.mat');
663                 if FileName ~= 0
664                     load([PathName,FileName]);
665                 end
666
667             case 18
668
669                 %% Load previous setting
670                 [FileName,PathName,~] = uigetfile('setting.mat');
671                 if FileName ~= 0
672                     load([PathName,FileName]);
673                 end
674
675              case 19
676
677                 %% close all figures
678                 close all;
679
680              case 20
681
682                 %% Load default setting
683                 loaddefault = input('16. Loading default values will
    reset all current setting, data, and analyses? (Y/N) [default: N]','
    s');
684                 if isempty(loaddefault)
685                     loaddefault = 'N';
686                 end
687
688                 if loaddefault == 'Y';
689                     defaultvalues;
690                 end
691
692              case 21
```

```
693
694                 %% Save data and setting
695                 vecsave = input('Save data and setting? (Y/N) [default:
     N]','s');
696                 if isempty(vecsave)
697                     vecsave = 'N';
698                 end
699
700                 if vecsave ~= 'N'
701                     uisave({'vec','dark','iso','not_ana','w','h','d','
    gridsize','total'},'data');
702                     uisave({'min_grid','max_grid','rselect','filsize','
    sig','th_up','th_low',...
703                         'gridsize','dark_th','iso_th','nint','dsize'},'
    setting');
704                 end
705
706             case 22
707
708                 while 1
709                     new_com = input('''!q'' to exit command mode >>','s
    ');
710                     if strcmp(new_com,'!q')
711                         break
712                     end
713                     if strcmp(new_com,'fiber3D')
714                         break
715                     end
716                     try
717                         eval(new_com);
718                     catch err
719                         fprintf(horzcat(err.message,'\n'));
720                     end
```

```
                end

            otherwise

                fprintf('Invalid selection! Press any key to continue.\
    n');
                pause
        end

     else
        if option == 'q'
            fprintf('Exit fiber3D. \n');
            vec = 0;
            f3D_avail = 1;
            return;
        else
            if ~isempty(opt) && sum(not_avail==opt) > 0
                fprintf('Currently unavailable selection! Press any key
     to continue.\n');
                pause
            else
                fprintf('Invalid selection! Press any key to continue.\
    n');
                pause
            end
        end
     end

end
```

Listing 4.4: Quantification of 3D Fiber Orientation

```
% setUP
name3D = 'control_d7_g24_notreally';
```

```
gridsize = 16;
%%
ang = double(ang);
ang(ang == 1000) = NaN;
%% Calculations
dimx = size(ang,1);
dimy = size(ang,2);
dimz = size(ang,3);

% update
rn = 1;
rnz = 1;
sizeby2  = gridsize / rn;
sizeby2z = gridsize / rnz;

% Synthetic sampling coordinates
x = 1:sizeby2:dimx*rn*sizeby2;
y = 1:sizeby2:dimy*rn*sizeby2;
z = 1:sizeby2z:dimz*rnz*sizeby2z;
[X,Y,Z] = meshgrid(x,y,z);

% Vector field from angles
u = sind(ang(:,:,:,1)) .* cosd(ang(:,:,:,2));
v = -cosd(ang(:,:,:,1)) .* cosd(ang(:,:,:,2));
w = sind(ang(:,:,:,2));
u = repelem(u, rn, rn, rnz);
v = repelem(v, rn, rn, rnz);
w = repelem(w, rn, rn, rnz);

% optional start grid (kept as in original)
[startX,startY,startZ] = meshgrid(1:gridsize:dimx*gridsize, 1:gridsize:
    dimy*gridsize, 1:gridsize:dimz*gridsize);
a = 2 + 2;
```

```
%% Energy and Enstrophy
%% Finite Difference

sideSize_x = dimx;
sideSize_y = dimy;
sideSize_z = dimz;

h1 = 1/sideSize_x; % delta x
h2 = 1/sideSize_y; % delta y
h3 = 1/sideSize_z; % delta z

% Prepare mesh for continuum-based FD (0..1 domain)
x = 0:h1:1; y = 0:h2:1; z = 0:h3:1;
[X,Y,Z] = meshgrid(x,y,z);

% Field components assigned for FD
f1 = u;
f2 = v;
f3 = w;

% --- FINITE DIFFERENCES ---
% NOTE: use consistent dimension indexing:
%   x-direction -> first array dimension (rows)
%   y-direction -> second array dimension (cols)
%   z-direction -> third array dimension (pages)

% d()/dx  (difference along first dimension)
FXf1 = (f1(2:end,:,:)-f1(1:end-1,:,:)) / h1;   % forward diff
FXf2 = (f2(2:end,:,:)-f2(1:end-1,:,:)) / h1;
FXf3 = (f3(2:end,:,:)-f3(1:end-1,:,:)) / h1;
% backward difference at boundary
BXf1 = (f1(end,:,:)-f1(end-1,:,:)) / h1;
BXf2 = (f2(end,:,:)-f2(end-1,:,:)) / h1;
```

```
BXf3 = (f3(end,:,:)-f3(end-1,:,:)) / h1;
% reassemble to same size as original along dim1
F1_x = cat(1, FXf1, BXf1);
F2_x = cat(1, FXf2, BXf2);
F3_x = cat(1, FXf3, BXf3);

% d()/dy  (difference along second dimension)
FYf1 = (f1(:,2:end,:)-f1(:,1:end-1,:)) / h2;
FYf2 = (f2(:,2:end,:)-f2(:,1:end-1,:)) / h2;
FYf3 = (f3(:,2:end,:)-f3(:,1:end-1,:)) / h2;
BYf1 = (f1(:,end,:)-f1(:,end-1,:)) / h2;
BYf2 = (f2(:,end,:)-f2(:,end-1,:)) / h2;
BYf3 = (f3(:,end,:)-f3(:,end-1,:)) / h2;
% reassemble to same size as original along dim2
F1_y = cat(2, FYf1, BYf1);
F2_y = cat(2, FYf2, BYf2);
F3_y = cat(2, FYf3, BYf3);

% d()/dz  (difference along third dimension)
FZf1 = (f1(:,:,2:end)-f1(:,:,1:end-1)) / h3;
FZf2 = (f2(:,:,2:end)-f2(:,:,1:end-1)) / h3;
FZf3 = (f3(:,:,2:end)-f3(:,:,1:end-1)) / h3;
BZf1 = (f1(:,:,end)-f1(:,:,end-1)) / h3;
BZf2 = (f2(:,:,end)-f2(:,:,end-1)) / h3;
BZf3 = (f3(:,:,end)-f3(:,:,end-1)) / h3;
% reassemble to same size as original along dim3
F1_z = cat(3, FZf1, BZf1);
F2_z = cat(3, FZf2, BZf2);
F3_z = cat(3, FZf3, BZf3);

% --- Curl components (keep separate arrays) ---
% (   F   ) = [ Cx, Cy, Cz ] with
%   Cx = dF3/dy - dF2/dz
```

```
%    Cy = -dF3/dx + dF1/dz
%    Cz = dF1/dx - dF2/dy
Cx = F3_y - F2_z;
Cy = -F3_x + F1_z;
Cz = F1_x - F2_y;

% Energy (sum of squared gradients)
F12xyz2 = F1_x.^2 + F2_x.^2 + F3_x.^2 + F1_y.^2 + F2_y.^2 + F3_y.^2 +
    F1_z.^2 + F2_z.^2 + F3_z.^2;
FD_K3Energy = 0.5 * h1 * h2 * h3 * sum(F12xyz2(:), 'omitnan');

% Enstrophy (sum of squared curl components)
FD_curl_sq = Cx.^2 + Cy.^2 + Cz.^2;
FD_Enstrophy = h1 * h2 * h3 * sum(FD_curl_sq(:), 'omitnan');

% Save matlab (commented)
% save(name3D + '.mat', 'color2','b','r','X','Y','Z','u','v','w')

%% Sphere / save angles
ang_z  = ang(:,:,:,2);
ang_xy = ang(:,:,:,1);
ang_z  = ang_z(:);
ang_xy = ang_xy(:);
ang_xyz = [ang_xy ang_z];
name_xyz = append(name3D, '_angxyz.csv');
writematrix(ang_xyz, name_xyz);

%% Figure 1 Vector Field
figure(2)
col2 = "#b4de2c";
col1 = "#30668d";

% Preallocate color vector and handle NaNs
```

```
color1000 = strings(numel(w),1);
idx_pos = w(:) > 0;
idx_nan = isnan(w(:));
color1000(idx_pos)  = col1;
color1000(~idx_pos) = col2;
color1000(idx_nan)  = col2; % assign something to NaNs (pie adjusts later)

color2 = reshape(color1000, size(w));

% Counting (exclude NaNs in w when computing opposing count)
b = sum(color2(:) == col1);
r = sum(color2(:) == col2) - nnz(isnan(w));
color = [b r];
figure(2); h2 = pie(color);
set(h2(1), 'FaceColor', col1, 'EdgeColor', col1);
set(h2(3), 'FaceColor', col2, 'EdgeColor', col2);

vfc_name = append(name3D, '_color.png');
saveas(gcf, vfc_name);

%% VF plotting (quiver3)
figure(3)
grid on; box on; hold on;

% Looping for per-voxel color (keeps your original approach)
for i = 1:size(Z,1)
    for j = 1:size(Z,2)
        for k = 1:size(Z,3)
            quiver3(X(i,j,k), Y(i,j,k), Z(i,j,k), u(i,j,k), v(i,j,k), w(i,j,k), ...
                'AutoScaleFactor', 2, 'Color', color2(i,j,k), 'LineWidth', 1.7, ...
```

```
                'AlignVertexCenters', 'on', 'ShowArrowHead', 'on', '
    MaxHeadSize', 1, 'AutoScale', 'on');
        end
    end
end

plt = gca;
xlabel('x [\mum]'); ylabel('y [\mum]'); zlabel('z [\mum]');

% Ticks (kept your values but could be automated)
set(gca, 'xtick', 0:200:dimx*rn*sizeby2);
set(gca, 'ytick', 0:200:dimy*rn*sizeby2);
set(gca, 'ztick', 0:15:dimz*rnz*sizeby2);
view(237,40); box on;

set(gca, 'linewidth', 0.01);
set(plt, 'FontName', 'Arial');
set(plt, 'fontweight', 'bold', 'fontsize', 17);

vf_name = append(name3D, '_VF_7q.png');
print(gcf, vf_name, '-dpng', '-r300');

%% Figure 2 Streamlines
xlabel('x [\mum]'); ylabel('y [\mum]'); zlabel('z [\mum]');

% Generate streamlines using seeding at grid points
verts = stream3(X, Y, Z, u, v, w, X, Y, Z);
str = streamline(verts);

view(3);
set(str, 'LineWidth', 1.2);
set(str, 'Color', 'r'); % initial color; will recolor below
axis tight;
```

```
plt2 = gca;
set(plt2, 'fontweight', 'bold', 'fontsize', 20);

set(gca, 'xtick', 0:200:dimx*rn*sizeby2);
set(gca, 'ytick', 0:200:dimy*rn*sizeby2);
set(gca, 'ztick', 0:15:dimz*rn*sizeby2);

% Color streamlines with a colormap
if numel(str) > 0
    colorstring = colormap(parula(numel(str)));
    for i = 1:numel(str)
        set(str(i), 'Color', colorstring(i,:));
    end
end

str_name = append(name3D, '_str.png');
saveas(gcf, str_name);

%% Figure 3 Tortuosity
% Calculation of tortuosity from streamlines
Xdata = get(str, 'Xdata');
Ydata = get(str, 'Ydata');
Zdata = get(str, 'Zdata');

% Ensure empty or single-point streamlines turned into NaNs
for ind = 1:length(Xdata)
    if iscell(Xdata) && size(Xdata{ind},2) == 1
        Xdata{ind} = NaN;
        Ydata{ind} = NaN;
        Zdata{ind} = NaN;
    end
end

```

```
Tor = cell(length(Xdata),1);
FTor = cell(length(Xdata),1);
for ind = 1:length(Xdata)
    Bx = rmmissing(Xdata{ind});
    By = rmmissing(Ydata{ind});
    Bz = rmmissing(Zdata{ind});
    if numel(Bx) > 1
        % end-to-end distance
        d = sqrt( (Bx(end)-Bx(1))^2 + (By(end)-By(1))^2 + (Bz(end)-Bz(1))^2 );
        % trajectory length
        tt = 0;
        for in = 1:(numel(Bx)-1)
            tr = sqrt( (Bx(in+1)-Bx(in))^2 + (By(in+1)-By(in))^2 + (Bz(in+1)-Bz(in))^2 );
            tt = tt + tr;
        end
        Tor{ind} = tt;
        if d > 0
            FTor{ind} = round(Tor{ind} / d, 2);
        else
            FTor{ind} = NaN; % degenerate streamline (zero end-to-end distance)
        end
    else
        FTor{ind} = NaN;
    end
end

All = FTor(:);
hist3 = cell2mat(All);
hist = hist3(~isnan(hist3));
name_hist = append(name3D, '_Tor.csv');
```

```
writematrix(hist, name_hist);

% Optional: histogram plotting commented (kept from original)
% figure
% histogram(hist,100,'Normalization','probability')
% xlabel('\tau','FontWeight','bold')
% ylabel('Frequency [a. u.]','FontWeight','bold')
% set(gca,'FontSize',20)
```

Listing 4.5: 3D Fluid-Inspired Fiber Analysis (FIFA) Metrics

```
"""Probe Lab

Nicholas Gaitanis
7/11/2022
"""

# File: vector_correct_3D

from nbformat import read
import networkx as nx
from matplotlib import mlab, pyplot as plt
import pandas as pd
import numpy as np
import math
import scipy.io as sio
import random

def network_plot_3D(G,angle):
    # https://www.idtools.com.au/3d-network-graphs-with-python-and-the-mplot3d-toolkit/
    # Get node positions
    pos = nx.get_node_attributes(G, 'pos')

```

```
    # Get the maximum number of edges adjacent to a single node
    edge_max = max([G.degree(i) for i in pos])

    # 3D network plot
    with plt.style.context(('ggplot')):

        fig = plt.figure(figsize=(10,7))
        ax = fig.add_subplot(111, projection="3d")

        # Loop on the pos dictionary to extract the x,y,z coordinates
    of each node
        for value in pos:
            xi = value[0]
            yi = value[1]
            zi = value[2]

            # Scatter plot
            ax.scatter(xi, yi, zi, c = 'green', s=20+20*G.degree(value)
    , edgecolors='k', alpha=0.7)

        # Loop on the list of edges to get the x,y,z, coordinates of
    the connected nodes
        # Those two points are the extrema of the line to be plotted
        for i,j in enumerate(G.edges()):

            x = np.array((pos[j[0]][0], pos[j[1]][0]))
            y = np.array((pos[j[0]][1], pos[j[1]][1]))
            z = np.array((pos[j[0]][2], pos[j[1]][2]))

        # Plot the connecting lines
            ax.plot(x, y, z, c='black', alpha=0.5)

    # Set the initial view
```

```
53      ax.view_init(30, angle)
54
55      # Hide the axes
56      ax.set_axis_off()
57      plt.show()
58
59  def init_graph(angles,avoid):
60      """A method to initilaize a 3D grid graph. This graph will be used
        to access neighbors of existing nodes
61      There is a lot of repeating code in this section. However, there
        are many diagonal angles to establish.
62      In the future this should be rewritten to be more consice.
63
64      Args:
65          angles: a list structure for the size of the graph. Either phi
        or theta can be used assuming they are similar
66          x, y, and z: sizes of the graph
67      """
68      angles = angles[:,:,:,0]
69      x = len(angles)
70      y = len(angles[0])
71      z = len(angles[0][0])
72      G = nx.Graph()
73      G.add_node((i,j,k) for i in range(x) for j in range(y) for k in
        range(z))
74
75      for i in range(x):
76          for j in range(y):
77              for k in range(z):
78                  G.add_node((i,j,k),pos=(i,j,k))
79
80      # ADD EDGES TO THE GRID GRAPH ON DIAGONALS XY PLANE
81      G.add_edges_from([((i, j, k), (i+1, j+1, k)) \
```

```
82          for i in range(x-1) for j in range(y-1) for k in range(z) \
83          if not (angles[i][j][k] == avoid or angles[i+1][j+1][k] ==
    avoid)] + \
84          [((i+1, j, k), (i, j+1, k)) for i in range(x-1) for j in range(
    y-1) for k in range(z)\
85          if not (angles[i+1][j][k] == avoid or angles[i][j+1][k] ==
    avoid)], weight=1)
86
87      # ADD EDGES TO THE GRID GRAPH ON DIAGONALS YZ PLANE
88      G.add_edges_from([((i, j, k), (i, j+1, k+1)) \
89          for i in range(x) for j in range(y-1) for k in range(z-1) \
90          if not (angles[i][j][k] == avoid or angles[i][j+1][k+1] ==
    avoid)] + \
91          [((i, j, k+1), (i, j+1, k)) for i in range(x) for j in range(y
    -1) for k in range(z-1)\
92          if not (angles[i][j][k+1] == avoid or angles[i][j+1][k] ==
    avoid)], weight=1)
93
94      # ADD EDGES TO THE GRID GRAPH ON DIAGONALS XZ PLANE
95      G.add_edges_from([((i, j, k), (i+1, j, k+1)) \
96          for i in range(x-1) for j in range(y) for k in range(z-1) \
97          if not (angles[i][j][k] == avoid or angles[i+1][j][k+1] ==
    avoid)] + \
98          [((i, j, k+1), (i+1, j, k)) for i in range(x-1) for j in range(
    y) for k in range(z-1)\
99          if not (angles[i][j][k+1] == avoid or angles[i+1][j][k] ==
    avoid)], weight=1)
100
101     # ADD EDGES TO THE GRID GRAPH ON DIAGONALS FOR XYZ VOLUME
102     G.add_edges_from([((i, j, k), (i+1, j+1, k+1)) \
103         for i in range(x-1) for j in range(y-1) for k in range(z-1) \
104         if not (angles[i][j][k] == avoid or angles[i+1][j+1][k+1] ==
    avoid)] + \
```

```
105         [((i, j, k+1), (i+1, j+1, k)) for i in range(x-1) for j in
    range(y-1) for k in range(z-1)\
106         if not (angles[i][j][k+1] == avoid or angles[i+1][j+1][k] ==
    avoid)] + \
107         [((i, j+1, k), (i+1, j, k+1)) for i in range(x-1) for j in
    range(y-1) for k in range(z-1)\
108         if not (angles[i][j+1][k] == avoid or angles[i+1][j][k+1] ==
    avoid)] + \
109         [((i+1, j, k), (i, j+1, k+1)) for i in range(x-1) for j in
    range(y-1) for k in range(z-1)\
110         if not (angles[i+1][j][k] == avoid or angles[i][j+1][k+1] ==
    avoid)], weight=1)
111
112     # ADD EDGES TO THE GRID GRAPH FOR ADJACENT LAYERS
113     G.add_edges_from([((i, j, k), (i+1, j, k)) \
114         for i in range(x-1) for j in range(y) for k in range(z) \
115         if not (angles[i][j][k] == avoid or angles[i+1][j][k] == avoid)
    ] + \
116         [((i, j, k), (i, j+1, k)) for i in range(x) for j in range(y-1)
     for k in range(z)\
117         if not (angles[i][j][k] == avoid or angles[i][j+1][k] == avoid)
    ] + \
118         [((i, j, k), (i, j, k+1)) for i in range(x) for j in range(y)
    for k in range(z-1)\
119         if not (angles[i][j][k+1] == avoid or angles[i][j][k] == avoid)
    ], weight=1)
120
121     return G
122
123 def bfs_traversal(G,theta,phi,start):
124     """This method is a breadth first search traversal for a 3D network
    . The network will change neighbors of phi and theta.
125     This network will determine which combination of phi and theta
```

```
    gives the smallest distance across a sphere.
126  Note that theta can only be changed by 180 degrees while phi can
    change in sign.
127
128  This method works as long as theta and phi are numpy arrays!!
129
130  Args:
131      angles: a list structure
132      G: the graph structure
133      start: coordinate location of first node to start on
134  Returns:
135      theta, phi: corrected anglular matrices for spherical
    coordinates
136  """
137  s = dict(nx.bfs_successors(G,start))
138  visited = dict.fromkeys(s.keys(),False)  # value is visted
139
140  queue = []
141  queue.append(list(s.keys())[0])
142
143  visited[list(s.keys())[0]] = True
144
145  while queue:
146      # Dequeue a vertex from queue
147      p = queue.pop(0)
148      x_arr, y_arr, z_arr = p
149      itheta = np.radians(theta[x_arr][y_arr][z_arr])
150      iphi = np.radians(phi[x_arr][y_arr][z_arr])
151      # convert to cartesian coordinate system
152      x1 = np.sin(iphi)*np.cos(itheta)
153      y1 = np.sin(iphi)*np.sin(itheta)
154      z1 = np.cos(iphi)
155
```

```
156         # Get all adjacent nodes
157         for i in s[p]:
158 
159             # Get neihboring angles
160             x_arrn, y_arrn, z_arrn = i
161             ftheta = np.radians(theta[x_arrn][y_arrn][z_arrn])
162             fphi = np.radians(phi[x_arrn][y_arrn][z_arrn])
163             # convert to cartesian coordinate system
164             x2 = np.sin(fphi)*np.cos(ftheta)
165             y2 = np.sin(fphi)*np.sin(ftheta)
166             z2 = np.cos(fphi)
167 
168             # Get alternate neighboring angles
169             altftheta = np.radians((theta[x_arrn][y_arrn][z_arrn] +
    180) % 360)
170             altfphi = np.radians(180 - phi[x_arrn][y_arrn][z_arrn])
171             # Convert to cartesian coordinate system
172             x3 = np.sin(altfphi)*np.cos(altftheta)
173             y3 = np.sin(altfphi)*np.sin(altftheta)
174             z3 = np.cos(altfphi)
175 
176             # Find the distance between two points on a unit sphere
177             del1 = 2*np.arcsin(math.sqrt((x1-x2)**2 + (y1-y2)**2 + (z1-
    z2)**2)/2)
178             del2 = 2*np.arcsin(math.sqrt((x1-x3)**2 + (y1-y3)**2 + (z1-
    z3)**2)/2)
179 
180             # Is the alternate distance less?
181             if del2 < del1:
182                 theta[x_arrn][y_arrn][z_arrn] = np.degrees(altftheta)
183                 phi[x_arrn][y_arrn][z_arrn] = np.degrees(altfphi)
184 
185             # Mark the traversal as 'visted'
```

```
            if i in s:
                if visited[i] == False:
                    queue.append(i)
                    visited[i] = True # Mark and enqueue it
    return theta, phi

def plot_3d_quivers(theta,phi,mask,blocks):
    """The following method will plot the angles of theta and phi on a 3D plot

    Args:
        theta, phi: numpy arrays for spherical coordinates
    """

    fig = plt.figure()
    ax = fig.add_subplot(projection='3d')
    x, y, z = np.meshgrid(range(len(theta[0])), range(len(theta)), range(len(theta[0][0])))

    u = np.sin(np.radians(phi))*np.cos(np.radians(theta))
    v = np.sin(np.radians(phi))*np.sin(np.radians(theta))
    w = np.cos(np.radians(phi))

    # The colormap for x, y, z, is RGB

    # Color by azimuthal angle
    c = np.arctan2(v, u)
    # Flatten and normalize
    c = (c.ravel() - c.min()) / c.ptp()
    # Repeat for each body line and two head lines
    c = np.concatenate((c, np.repeat(c, 2)))
    # Colormap
    c = plt.cm.hsv(c)
```

```
217
218
219     ax.quiver(x[mask],y[mask],z[mask],u[mask],v[mask],w[mask],length =
        0.5, colors = c)
220
221     #ax.set_box_aspect((np.ptp(x), np.ptp(y), np.ptp(z)))
222     set_axes_equal(ax)
223     plt.tight_layout()
224     plt.show(block = blocks)
225
226 def set_axes_equal(ax):
227     """Make axes of 3D plot have equal scale so that spheres appear as
        spheres,
228     cubes as cubes, etc..  This is one possible solution to Matplotlib'
        s
229     ax.set_aspect('equal') and ax.axis('equal') not working for 3D.
230
231     Input
232       ax: a matplotlib axis, e.g., as output from plt.gca().
233     """
234
235     x_limits = ax.get_xlim3d()
236     y_limits = ax.get_ylim3d()
237     z_limits = ax.get_zlim3d()
238
239     x_range = abs(x_limits[1] - x_limits[0])
240     x_middle = np.mean(x_limits)
241     y_range = abs(y_limits[1] - y_limits[0])
242     y_middle = np.mean(y_limits)
243     z_range = abs(z_limits[1] - z_limits[0])
244     z_middle = np.mean(z_limits)
245
246     # The plot bounding box is a sphere in the sense of the infinity
```

```
247     # norm, hence I call half the max range the plot radius.
248     plot_radius = 0.5*max([x_range, y_range, z_range])
249 
250     ax.set_xlim3d([x_middle - plot_radius, x_middle + plot_radius])
251     ax.set_ylim3d([y_middle - plot_radius, y_middle + plot_radius])
252     ax.set_zlim3d([z_middle - plot_radius, z_middle + plot_radius])
253 
254 def test_case1():
255     x, y, z = np.meshgrid(np.linspace(-1,1,5),np.linspace(-1,1,5),np.
    linspace(-1,1,5))
256 
257     u = x + np.cos(4*x) + 3
258     v = np.sin(4*x) - np.sin(2*y)
259     w = -z
260 
261     #ax = plt.figure().add_subplot(projection='3d')
262     #ax.quiver(x, y, z, u, v, w,  length = 0.05)
263     #plt.show(block = False)
264 
265     theta = np.degrees(np.arctan2((v-y),(u - x)))
266     phi = np.degrees(np.arccos((w-z)/np.sqrt((u-x)**2 + (v-y)**2 + (w-z
    )**2)))
267 
268     plot_3d_quivers(theta,phi,False)
269 
270     for i in range(len(theta)):
271         for j in range(len(theta[0])):
272             for k in range(len(theta[0][0])):
273                 if theta[i][j][k] < 0:
274                     theta[i][j][k] = (theta[i][j][k] + 180) % 360
275                     phi[i][j][k] = 180 - phi[i][j][k]
276     return theta, phi
277 
```

```
278 def test_case2():
279     x, y, z = np.mgrid[0:1:20j, 0:1:20j, 0:1:20j]
280 
281     u = np.sin(np.pi*x) * np.cos(np.pi*z)
282     v = -2*np.sin(np.pi*y) * np.cos(2*np.pi*z)
283     w = np.cos(np.pi*x)*np.sin(np.pi*z) + np.cos(np.pi*y)*np.sin(2*np.
    pi*z)
284 
285     theta = np.degrees(np.arctan2((v-y),(u - x)))
286     phi = np.degrees(np.arccos((w-z)/np.sqrt((u-x)**2 + (v-y)**2 + (w-z
    )**2)))
287 
288     plot_3d_quivers(theta,phi,False)
289 
290     return theta, phi
291 
292 def read_to_list(dir):
293     df = pd.read_csv(dir,header = None)
294     l = df.values.tolist()
295     return np.asarray(l)
296 
297 def correct_from_matlab(theta,phi,start):
298     """The following function can be ran from matlab to correct for
299     3D vector data
300 
301     Input
302       Theta: A 3D matrix of theta values
303       Phi: A 3D matrix of phi values
304       start: Starting location for the traversal as a triple
305     """
306     G = init_graph(theta)
307     theta, phi = bfs_traversal(G,theta,phi,start)
308     return theta,phi
```

```
309
310 def aniso_list(arr,iso_val):
311     mask = np.full(np.shape(arr), True)
312     mask[arr == 1000] = False
313     indices = []
314     for i in range(len(arr)):
315         for j in range(len(arr[0])):
316             for k in range(len(arr[0][0])):
317                 if mask[i][j][k] == 1:
318                     indices.append((i,j,k))
319     return mask, indices
320
321 def dist_traversal(indices,theta,phi):
322     """This method traverses indices with angular values.
323     It prioritizes changing angular values by caluclating distances
    between indices in a 3D space.
324
325     Args:
326         indices: a list of indices with angular values
327         theta: values of theta
328         phi: values of phi
329     """
330     distances = np.array([])
331
332     for i in range(1,len(indices)):
333         d = np.sqrt((indices[i][1] - indices[i-1][1])**2 + (indices[i
    ][0] - indices[i-1][0])**2 + (indices[i][2] - indices[i-1][2])**2)
334         distances = np.append(distances,d)
335
336     sorted_by_arg = np.argsort(distances)
337     indices = np.array(indices)[sorted_by_arg]
338
339     for i in range(1,len(indices)):
```

```
340         const_theta = theta[indices[i-1][1],indices[i-1][0], indices[i
    -1][2]]
341         const_phi = phi[indices[i-1][1],indices[i-1][0], indices[i
    -1][2]]
342 
343         const_x = np.sin(const_phi)*np.cos(const_theta)
344         const_y = np.sin(const_phi)*np.sin(const_theta)
345         const_z = np.cos(const_phi)
346 
347         comp_theta = theta[indices[i][1],indices[i][0], indices[i][2]]
348         comp_phi = phi[indices[i][1],indices[i][0], indices[i][2]]
349 
350         comp_x = np.sin(comp_phi)*np.cos(comp_theta)
351         comp_y = np.sin(comp_phi)*np.sin(comp_theta)
352         comp_z = np.cos(comp_phi)
353 
354 def generate_mask(arr,val=1000):
355     arr = arr[:,:,:,0]                  # Select From Either Theta or
    Phi
356     mask = np.full(np.shape(arr), True)
357     mask[arr == val] = False
358     indices = []
359     for i in range(len(arr)):
360         for j in range(len(arr[0])):
361             for k in range(len(arr[0][0])):
362                 if mask[i][j][k] == 1:
363                     indices.append((i,j,k))
364     return mask, indices
365 
366 def main():
367     # Preliminary Random Constants
368     rand = 11
369     random.seed(rand)
```

```
    np.set_printoptions(threshold=np.inf)

    # Load in matlab data
    #mat_fname = 'matlab_tgfb_d21_v2.mat' # File to be imported
    #mat_fname='matlab_data_tgfb_d7_i0999_d0030_g24_v2.mat'
    mat_fname ='matlab_UPenn_300um_liver_g24_d0.062_0.999i.mat'
    mat_contents = sio.loadmat(mat_fname)
    angular_vals = np.array(mat_contents[list(mat_contents.keys())
    [-1]]).astype(np.int32) # important that we cast the correct type
    mask, indices = generate_mask(angular_vals) # generate mask

    # Vizualize Direction
    theta = angular_vals[:,:,:,0]
    phi = angular_vals[:,:,:,1]
    phi[mask] = phi[mask] - 90
    plot_3d_quivers(theta,phi,mask,True)

    # Correct Values
    G = init_graph(angular_vals,1000)
    start_index = indices[random.randint(0,len(indices)-1)]
    theta, phi = bfs_traversal(G,theta,phi,start_index)
    plot_3d_quivers(theta,phi,mask,True)

    # Export Values
    phi[mask] = phi[mask] + 90
    angular_vals2 =  np.stack((theta, phi), axis = 3)
    mat_contents[list(mat_contents.keys())[-1]] = angular_vals2
    # sio.savemat('corrected_matlab_tgfb_d21_v2.mat',mat_contents)
    sio.savemat('corrected_matlab_UPenn_300um_liver_g24_d0.062_0.999i.
    mat',mat_contents)

if __name__ == '__main__':
```

```
400     main()
```

Listing 4.6: 3D FIFA Correction Algorithm

```
1
2  %% Step 1: Load a single image (for testing threshold)
3  % Example paths (comment/uncomment as needed):
4  % I = 'R:\...\Control\13+b 2.0 d7 control a3 2.0_0001.tif';
5  % I = 'R:\...\TGFB\13+b 2.0 d21 tgfb hv5000005.tif';
6  I = 'R:\ENG_PROBELab_Shared\group\Adriana_files\Digital Volumes\Tendon
       \100um_SHG\100um_normal-20241113T231214Z-001\100um_normal\Image0004.
       tif.frames\Image0004_C002Z001.tif';
7
8  Ib = imread(I);    % read grayscale TIFF slice
9
10 %% Step 2: Thresholding (Triangle method + Otsu comparison)
11 [lehisto, x] = imhist(Ib);                          % histogram
12 [level] = triangle_th(lehisto,256);                 % triangle threshold
13 I_bw = imbinarize(Ib, level);                       % binary collagen mask
14
15 % Compare with Otsu's method
16 I_otsu = imbinarize(Ib, graythresh(Ib));
17
18 %% Step 3: Area calculation for one slice
19
20 Total_area_um2 = 508.8 * 508.8;                     % total FOV area (um
       ^2)
21 px_um2 = (0.7950004)^2;                             % calibration (um^2/px
       )
22 Col_area_px = bwarea(I_bw);                         % collagen area in px
23 Col_area_um2 = Col_area_px * px_um2;                % collagen area in um
       ^2
24 Per = Col_area_um2 / Total_area_um2;                % area fraction
25
```

```
%% Step 4: Visualization for single slice
figure;
subplot(2,3,1); imshow(Ib); axis off; title('Input image');
subplot(2,3,2); imshow(I_bw); axis off; title('Triangle method');
subplot(2,3,3); imshow(I_otsu); axis off; title('Otsu (graythresh)');
subplot(2,3,[4:6]); bar(x,lehisto); xlim([0 255]);
line([79 79],[0 max(ylim)],'Color','b','LineWidth',2);   % triangle
    thresh
line([126 126],[0 max(ylim)],'Color','g','LineWidth',2); % Otsu thresh
legend('Histogram','Triangle Threshold','Otsu Threshold');


% Batch Processing of Z-Stack


% Example filebase for image stack
filebase = 'R:\ENG_PROBELab_Shared\group\Adriana_files\Digital Volumes\
    Tendon\100um_SHG\100um_normal-20241113T231214Z-001\100um_normal\
    Image0004.tif.frames\Image0004_C002Z0';

startFrame = 10;   % first slice index
endFrame   = 72;   % last slice index
stepSize   = 1;    % step between slices

% Calibration (update for your microscope)
px_um2 = (1/2.0121)^2;        % pixel area (um^2)
Total_area_um2 = 254.46^2;    % total area of one slice (um^2)
z_step_um = 1.0;              % thickness per slice (um) <-- UPDATE if
    known

Col_vol = zeros(1,endFrame);  % preallocate

for i = startFrame:stepSize:endFrame
```

```
    % Build filename
    filename = [filebase, num2str(i,'%2d'), '.tif'];
    Ib = imread(filename);

    % Threshold (triangle method)
    [lehisto,~] = imhist(Ib);
    [level] = triangle_th(lehisto,256);
    I_bw = imbinarize(Ib, level);

    %Collagen area in um2
    Col_area_px = bwarea(I_bw);
    Col_area_um2 = Col_area_px * px_um2;

    % Store slice area
    Col_vol(i) = Col_area_um2;
end

%% Step 5: Volume calculations
numSlices = (endFrame - startFrame + 1);
cube_vol = Total_area_um2 * numSlices * z_step_um;   % um^3
total_vol = sum(Col_vol, 'omitnan') * z_step_um;     % collagen volume
   (um^3)
per_vol = total_vol / cube_vol;                       % volume fraction

fprintf('Collagen Volume = %.2f um^2\n', total_vol);
fprintf('Total Cube Volume = %.2f um^2\n', cube_vol);
fprintf('Volume Fraction = %.4f\n', per_vol);


% Synthetic Volumetric Dataset Example

filebase = 'R:\ENG_PROBELab_Shared\group\Adriana_files\Digital Volumes\
   Slighly Wavy\circlinear1_7sin(x_15)\circlinear1_7sin(x_15)_';
```

```
startFrame = 1;
endFrame = 256;

Col_area_px = zeros(1,endFrame);

for i = startFrame:endFrame
    filename = [filebase, num2str(i,'%2d'), '.tif'];
    Ib = imread(filename);

    % Here images are already binary
    I_bw = Ib;

    % Collagen area in px
    Col_area_px(i) = bwarea(I_bw);
end

Total_area_px = size(Ib,1) * size(Ib,2);  % area in px^2
cube_vol = Total_area_px * endFrame;      % total cube volume (px^3)
total_vol = sum(Col_area_px);             % collagen volume (px^3)
per_vol = 100 * total_vol / cube_vol;     % %

fprintf('Synthetic Volume Fraction = %.2f %%\n', per_vol);
```

Listing 4.7: Collagen Volume Fraction Calculation from SHG Image Stacks