

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Biopolymer Networks: Image Analysis, Reconstruction and Modeling

*A thesis presented in partial fulfilment of the requirements
for the degree of*

*Doctor of Philosophy
in
Physics*

*at Massey University, Manawātū,
New Zealand*

Pablo Hernandez-Cerdan

2018



MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

It's not about the end, it's all about the path.

¡Buen camino!.

Abstract

The aim of this work is to extract the architecture of biopolymer networks from 3D images. This is motivated to further understand non-affine regimes, found during network formation and in low-density biopolymer networks, where the geometry of a network has a fundamental role in defining its mechanical properties.

Firstly, developed image analysis tools were extended to 3D and contributed to high-performance open-source libraries for image analysis. These developments in isotropic wavelets will help in extracting realistic networks by removing spurious noise generated during image acquisition.

Secondly, images of biopolymer gels from transmission electron microscopy (TEM), were used to reliably extract the network architecture. The imaged material was also studied with small-angle x-ray scattering (SAXS), and the comparison showed a strong agreement for network-size features.

Thirdly, spatial graphs were extracted from the image. A one-to-one map is provided between image and graph, keeping all the geometric information from the image. This then opened the door to using analytical tools from the complex networks field to characterize images.

Finally, statistical distributions extracted from three graph properties were used to reconstruct a completely in-silico network using a simulated annealing technique to generate new networks. This can be used as a computational exploration tool of how network behavior depends on network architecture.

Acknowledgements

A special thank to my supervisor Bill Williams whose support, openness, and vision of science is encouraging. As Brad would say, I will be forever indebted to your supervision and I add: and from all the stuff that I have learned from you beyond academia, specially your humane values and the love for dumplings. Thanks to Brad Mansel, Leif Lundin and Andrew Leis for the collaboration comparing scattering and imaging techniques, Allan Raudsepp for providing the confocal microscopy images for actin, and to all my group mates from the [Biophysics and Soft Matter group](#), for all the shared moments, and how easy they made my adaptation to a new country. Finally, I am grateful to the [MacDiarmid Institute](#) for constantly providing the enrichment opportunity to meet other peer students and researchers, and to the staff of the Institute of Fundamental Sciences for its kindness dealing with all sort of issues.

In the personal side, I would like to thank my family, from who I have inherited, directly and indirectly, all the principles I stand for and made me realize how privileged I am to have them. And Martica! Without you here in New Zealand this quest would have been impossible, or faster, who knows, but way less enjoyable for sure. It has been a pleasure habernos re-encontrado por kiwi-land. And even we got married, in English!!, crazy, but made me feel extremely fortunate. Thanks really for all your support.

This research was supported by the MacDiarmid Institute, the Riddet Institute, and Massey University, New Zealand.

Contents

Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	vii
List of Publications	x
Abbreviations	xi
1 Introduction	1
1.1 Soft matter and biopolymers	1
1.2 Semiflexible single chains: WLC model	2
1.3 Networks of semiflexible chains:	5
1.4 Bulk Rheology	8
1.4.1 Linear viscoelasticity	8
1.4.2 Nonlinear behaviour: strain stiffening	9
1.5 Network Structure	11
1.5.1 Mikado Networks	11
1.5.2 Lattice Models	12
1.5.3 Eight-chain Models	13
1.6 Motivation	14
2 Isotropic and Steerable Wavelets in N Dimensions. A multiresolution analysis framework	16
2.1 Motivation	16
2.2 Abstract	16
2.3 Wavelet Multiresolution Analysis	17
2.3.1 Introduction	17
2.3.2 Motivation: spatial and frequency resolution	18
2.3.3 Wavelet transformation	19
2.3.4 Wavelet Pyramid	21
2.3.5 Isotropic wavelets	22
2.4 Riesz Transform	25
2.4.1 Monogenic Signal	26

2.4.2	Generalized Riesz Transform	27
2.4.3	Generalized Steerable Framework	27
2.5	Implementation Details	28
2.5.1	Summary	28
2.5.2	Frequency Iterators	29
2.5.3	Wavelet Transform	30
2.5.4	Riesz Transform	31
2.5.5	Structure Tensor	31
2.5.6	Phase Analysis	32
2.6	A guided example:	32
2.6.1	Input in the frequency domain.	33
2.6.2	Choosing an isotropic wavelet	33
2.6.3	Forward / Analysis	33
2.6.4	Inverse / Reconstruction	34
2.6.5	PhaseAnalyzer	36
2.7	Conclusion and future work	41
2.8	Acknowledgements	42
3	Validation of Transmission Electron-Microscopy Imaging: Comparison with SAXS	43
3.1	Motivation	43
3.2	Abstract	43
3.3	Introduction	44
3.4	Materials and methods	45
3.5	Results and Discussion	48
3.5.1	Qualitative Structural Features	49
3.5.2	Consistency Between Techniques at Different Lengthscales	50
3.5.3	Image Processing	52
3.5.4	Extracting Persistence Lengths	54
3.6	Conclusions	55
4	Spatial Graph Extraction: Analysis of microscopy images to obtain network architecture	58
4.1	Motivation	58
4.2	Introduction	58
4.2.1	Notation	60
4.3	Denoising	61
4.4	Segmentation	62
4.4.1	Binarization	63
4.4.1.1	Region Growing Segmentation	64
4.4.2	Hole Filling	65
4.5	Skeletonization	66
4.5.1	Distance maps	68
4.5.2	Critical Kernels Framework	68
4.6	Spatial Graph Extraction	72
4.7	Statistical distributions of graph properties	74
4.8	Testing existing thinning algorithms	75

4.8.1	Fiber Extraction -FIRE-	75
4.8.1.1	Results in FIRE	77
4.8.2	Avizo: XSkeleton	79
4.8.2.1	Avizo results:	81
4.9	Application of the Network Analysis Pipeline to Different Biopolymer Networks.	84
4.9.1	Confocal Light Microscopy: Actin	84
4.9.2	TEM preparation	90
4.9.3	Transmission Electron Microscopy: Carrageenan	91
4.9.4	Transmission Electron Microscopy: Pectin	94
4.10	Conclusions	97
5	Reconstructing networks from statistical distributions	99
5.1	Motivation	99
5.2	Methods	99
5.3	Euclidean Graph Generation Algorithm	102
5.4	Results	103
6	Conclusions and Future Work	106
6.1	Scope of Thesis	106
6.2	Summary	106
6.3	Conclusions	108
6.4	Future work	109
6.4.1	Network Formation	109
6.4.2	Long time behaviour: network quakes and aging	109
6.4.3	ITKBoostGraph	110
6.4.4	Complex Networks Tools	110
6.4.5	Public Database	110
A	Publications and Open Source Contributions	111
A.1	Open Source Contributions	111
A.1.1	ITKIsotropicWavelets	111
A.1.2	Radial Intensity FFT	111
A.1.3	FFT Radial Intensity	112
A.1.4	DGtal: Critical Kernels Framework	112
A.1.5	Spatial Graph Extraction	113
A.1.6	Open Science	113
	Glossary	114
	References	116

List of Figures

1.1	Optical Tweezers	5
1.2	Force extension curve: CEWLC	6
1.3	Strain-stiffening in semiflexible polymers	9
1.4	Affine and non-affine deformations	10
1.5	Affine and non-affine phases	11
1.6	Mikado network used as a starting scaffold for dynamics simulations, from Ref. [31].	12
1.7	Unit cell of a face-centered-cubic lattice, from Ref. [40]	13
1.8	Eight-chain model under shear, from left to right: initial state, a rotation, and a simple shear, from Ref. Palmer and Boyce [30]	14
2.1	The changes over time/space of a signal are not captured by the FFT.	18
2.2	Simultaneous spatial and frequency resolution of different transformations. $\Delta t, \Delta f$	19
2.3	Forward 2.3a and Inverse wavelet 2.3b two-level pyramid with two high pass sub-bands.	21
2.4	Forward wavelet pyramid with a classic two level filter bank and only one sub-band, HP is the high pass filter, and LP low pass	22
2.5	Tiling of the frequency domain by isotropic wavelets when the dilation factor is 2. All the wavelets fulfill the conditions from Proposition 2.3. The mother wavelets are represented at $i = 0$	24
2.6	Shape of SubBands when HighPassSubBands = 5, sub-bands increase the frequency resolution, but can generate extra artifacts in the spatial domain due to the sharp frequency cut-offs.	25
2.7	The original image looks like a regular check-board, but it isn't. Pixels in the regions A and B have the same intensity value (129), however our vision system performs local phase analysis that allows us to treat A,B regions as different, keeping a global checker-board structure. 2.7b uses a non-linear map of intensity-color to enhance the irregular checker board structure.	37
2.8	(Results of the phase analysis (with soft threshold) for different number of scales in the wavelet pyramid. The input image is a checker-board of size 512×512).	38
2.9	Using six scales (Level: 6), results for different number of high frequency sub-bands.	39
2.10	The optical illusion generated by 2.10a the Hermann grid is generated by the local phase analysis of our vision. 2.10b is the result of a phase analysis with the Monogenic signal for a wavelet pyramid of 8 levels and only one high pass sub band. 2.10c is the same pyramid but with 10 high pass bands to increase the frequency resolution. Both results use Simoncelli wavelet, but there is not much difference using Held, or Vow mother wavelets.	40

2.11	Biopolymer network (actin) 3D image enhanced and denoised using phase analysis with the Monogenic signal, with a wavelet pyramid of 4 levels and 4 high pass sub-bands using Simoncelli isotropic wavelet.	41
3.1	The Graphical User Interface (GUI) of the image analysis software developed herein. A non-interactive interface is also provided for batch processing, along with python scripts for plotting and manipulation of the generated data. . . .	48
3.2	Montages of TEM images obtained from different polysaccharides, as described in the text.	49
3.3	SAXS results for the three samples shown in the TEM micrographs of figure 3.2, in the absence of TEM sample preparation. Black lines signify power law fitting, as is described in section 3.1 of the text.	50
3.4	Comparison of data obtained from TEM best practice and SAXS. Good agreement can be seen in the large network scale (at low q s). The shaded area shows the upper q range (smaller than 20 nm or equivalently 25 pixels). The top x-axis shows the corresponding real-space length scale, where $d = 2\pi/q$. .	53
3.5	Results obtained when applying two widely-used de-noising techniques: total variation regularization (TV) and multi-scale wavelet denoising using a BLS-GSM algorithm.	53
3.6	Visual comparison of the de-noised pectin TEM images (Fig 3.2a). (a) shows a region from the unmodified original image, (b): de-noised image using a total variation method [78] with $\lambda = 0.15$, (c) shows the same region after applying a multi-scale wavelet approach using the BLS-GSM algorithm [58] with $\sigma = 28.4$. Only high-frequency regions are affected by de-noising algorithms, as reflected in the generated I-q plots shown in Fig 3.5a.	54
3.7	A TEM micrograph with FibreApp tracking output shown in blue. The insert shows the internal contour length vs mean-square end-to-end distance calculated from the corresponding tracking parameters.	55
4.1	Visualization of a Spatial Graph (4.1a) and the corresponding adjacency list (4.1b). The graph is represented in blue, formed by nodes (voxels with id) and edges connecting them. The voxel positions (in brackets in 4.1a) of nodes and edge points are stored as labels in the adjacency list.	59
4.2	Pre-processing the image (actin): denoising with wavelets and total variation methods.	62
4.3	Different threshold values on networks	63
4.4	Thin image of a pectin network after skeletonization performed with the critical kernels framework (see subsection 4.9.4) and visualized with a 3D Viewer. The skeleton keeps the topology of the original object, is thin, centered, and robust against rough and noisy boundaries.	67
4.5	Distance map - Actin	68
4.6	Asymmetric thinning algorithm with persistence, from Ref. [91]	70
4.7	Asymmetric thinning of a volume with different parameters. (a): Ultimate skeleton, only voxels that conserve topology are kept. (b): keeps 1-isthmus as part of the skeleton. (c): 1-isthmus with persistence parameter $p = 10$	71
4.8	Skeletonization	72
4.9	Extracting the raw graph from the thin image and post processed to remove the largest edges between tri-connected nodes with degree 3. Numbers show the number of edges (degree) of each node.	74

4.10	Spatial Graph where vertices with degree 2 are deleted, and its position are stored as edge points in the new edges. Numbers in circles represent degree of each node.	74
4.11	FIRE algorithm - Find LMP	77
4.12	Fire: Step by step for $T_P = 0.12$	78
4.13	Distributions of length and degree in FIRE	79
4.14	Avizo image: Actin network visualization and workspace	80
4.15	Distributions of length, degree, and cosines with Avizo for actin with $T=30, P=5$	82
4.16	Distributions of length, degree, and cosines with Avizo for actin with $T=21, P=5$	83
4.17	Actin spatial graph extraction steps.	88
4.18	Statistical distributions (PDF) of actin network. Computed properties of the graph are represented in histograms with bins and normalized by area to obtain the PDF. Orange line shows the function with parameters obtained from a non-linear least squares fit to the data. Green lines show the same function with fixed parameters: 4.18a: mean degree (Z) of the network, 4.18c: mean (μ_l) and standard deviation (s_l) of logarithmic end-to-end distances, 4.18d mean (μ_l) and standard deviation (s_l) of logarithmic contour lengths. Histogram bins: end-to-end distances (30), direction cosines (21) contour lengths (30)	90
4.19	Potassium carrageenan spatial graph extraction steps.	93
4.20	Statistical distributions (PDF) of Potassium Carrageenan network. Computed properties of the graph are represented in histograms with bins and normalized by area to obtain the PDF. Orange line shows the function with parameters obtained from a non-linear least squares fit to the data. Green lines show the same function with fixed parameters: 4.20a: mean degree (Z) of the network, 4.20c: mean (μ_l) and standard deviation (s_l) of logarithmic end-to-end distances, 4.20d mean (μ_l) and standard deviation (s_l) of logarithmic contour lengths. Histogram bins: end-to-end distances (50), direction cosines (21) contour lengths (50)	94
4.21	Pectin spatial graph extraction steps.	96
4.22	Statistical distributions (PDF) of Pectin network. Computed properties of the graph are represented in histograms with bins and normalized by area to obtain the PDF. Orange line shows the function with parameters obtained from a non-linear least squares fit to the data. Green lines show the same function with fixed parameters: 4.22a: mean degree (Z) of the network, 4.22c: mean (μ_l) and standard deviation (s_l) of logarithmic end-to-end distances, 4.22d mean (μ_l) and standard deviation (s_l) of logarithmic contour lengths. Histogram bins: end-to-end distances (50), direction cosines (21) contour lengths (50)	97
5.1	Collagen distributions	100
5.2	Testing with theoretical collagen data	104
5.3	Collagen: comparing target and simulated distributions	105
6.1	Scheme of the thesis	108

List of Publications

- [1] Pablo Hernandez-Cerdan. “Isotropic and Steerable Wavelets in N Dimensions. A Multiresolution Analysis Framework for ITK.” In: *arXiv:1710.01103 [cs]* (Oct. 2017). arXiv: [1710.01103 \[cs\]](https://arxiv.org/abs/1710.01103).
- [2] Pablo Hernandez-Cerdan, Bradley W. Mansel, Andrew Leis, Leif Lundin, and Martin A.K. Williams. “Structural Analysis of Polysaccharide Networks by Transmission Electron Microscopy: Comparison with Small-Angle X-Ray Scattering.” In: *Biomacromolecules* (Jan. 2018). ISSN: 1525-7797. DOI: [10.1021/acs.biomac.7b01773](https://doi.org/10.1021/acs.biomac.7b01773).
- [3] Hina Shah, Pablo Hernandez, Francois Budin, Deepak Chittajallu, Jean-Baptiste Vimort, Rick Walter, André Mol, Asma Khan, and Beatriz Paniagua. “Automatic Quantification Framework to Detect Cracks in Teeth.” In: *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*. Vol. 10578. International Society for Optics and Photonics, Mar. 2018, 105781K. DOI: [10.1117/12.2293603](https://doi.org/10.1117/12.2293603).

Abbreviations

AFM	Atomic force microscopy.
BGL	Boost Graph Library. c++ library.
CEWLC	Clickable extensible wormlike chain.
DFT	Discrete Fourier Transform, representation of data in frequency domain.
DGtal	Digital Geometry Tools and Algorithms. c++ library.
EtE	End to end.
EWLC	Extensible wormlike chain.
FE	Force extension (curve).
FFT	Fast Fourier Transform, specific algorithm of DFT.
ITK	Insight ToolKit. Image Analysis c++ library specialized to work with ND images.
LAOS	Large amplitude oscillatory shear (rheology).
LMP	Local maximum point.
MRA	Multiresolution framework.
MSD	Mean square displacement.
NP	Nucleation point.
PME	Pectin-methylesterase.
SAXS	Small angle X-ray scattering .
SEM	Scanning electron microscopy.
TEM	Transmission electron microscopy.
TV	Total Variation (regularization, denoising method).
WLC	Wormlike chain.

1. Introduction

1.1 Soft matter and biopolymers

Soft matter, also known as complex fluids, is a subfield of condensed matter, comprising systems which organize at many different length scales, into many forms and classify as something in between the ordered solids and disordered liquids. Its conformation is heavily influenced by thermal fluctuations in the energy scale comparable to the room temperature. This characteristic allows conformational changes where complex behaviour can occur, life as the extreme example.

Soft matter includes liquids, colloids, polymers, foams, gels, granular materials, surfactants, liquids crystals and some biological materials. Pierre-Gilles de Gennes, who has been called the “founding father of soft matter” received the Nobel Prize in physics in 1991 “for discovering that methods developed for studying order phenomena in simple systems can be generalized to more complex forms of matter, in particular to liquid crystals and polymers”[1]

The biological soft materials span as well different length scales: from sugar chain structures to active gels inside cells. In the lowest scale the biopolymers are found, which are classified as polysaccharides (cellulose, pectin), polynucleotides (RNA, DNA) or polypeptides (proteins).

Most biopolymers are considered semi-flexible, something in between rigid rods and completely flexible (loose) chains. To formalize this distinction two parameters are introduced that characterize a polymer chain: the *persistence length* ℓ_p which is the typical length scale for the decay of tangent-tangent correlations, and the *contour length* L_c , which is defined as the maximum end-to-end distance of a linear polymer chain.

A chain is considered flexible when $\ell_p \ll L_c$, and rigid when the opposite holds. Completely flexible chains exhibit a purely entropic elastic response, and rigid filaments display no entropic,

but purely enthalpic response. Semi-flexible biopolymers have a similar magnitude of ℓ_p and L_c . This kind of filaments do not form loops or knots, but they are flexible enough to have thermal bending fluctuations[2]. They behave like rods at scales smaller than ℓ_p and like random coils in larger scales where the entropic behavior dominates.

1.2 Semiflexible single chains: WLC model

Single semiflexible chains can be modeled with great success with the *worm-like chain* (WLC) also known as the *Kratky-Porod* model [3, 4]. This model describes the single chain as an idealized curve that resists bending. Such bending deformations are described by the following Hamiltonian:

$$H_{bend} = \frac{\kappa}{2} \int ds \left| \frac{\partial \mathbf{t}}{\partial s} \right|^2$$

where $k \equiv \kappa_{\perp}$ is the bending modulus, \mathbf{t} is a unit tangent vector along the chain and s is the coordinate of the position along the backbone. In the WLC model, the persistence length is related with the bending modulus via: $\ell_p = 2\kappa_{\perp}/((d-1)k_B T)$ where d is the space dimensionality.

For nearly straight filaments $\frac{\partial \mathbf{t}}{\partial s}$ can be expressed via the transverse deviation $u(x)$ of the chain from its straight conformation. $\frac{\partial \mathbf{t}}{\partial s} = u''(x)$. If the chain is under a tensional force from one end (and the other end fixed), a term $H_{external} = -fL$ can be added to the Hamiltonian, with L being the end to end distance. The Hamiltonian with this force f_B in transverse coordinates is

$$H = \frac{1}{2} \int_0^{L_c} dx \left[\kappa |u''|^2 + f_B |u'|^2 \right] \quad (1.1)$$

where L_c is the contour length of the chain. The applicability of equation 1.1 must be questioned in some cases since it neglects excluded volume (steric repulsion effects) between the chain constituents completely [5]. However, for very stiff polymer these excluded volume considerations can be safely ignored.

Such a chain can respond to transverse and also to longitudinal forces by either bending or stretching/compressing. The force-extension (FE) relationship can be explored further,

decomposing u in Fourier series:

$$u(x) = \sum_q u_q \sin(qx) \quad (1.2)$$

with the wave vector $q = n\pi/L_c$. 1.1 can be rewritten as:

$$H = \frac{1}{2} \int_0^{L_c} dx \left[\kappa |u''|^2 + f_B |u'|^2 \right] = \frac{L_c}{4} \sum_q (\kappa q^4 + f_B q^2) u_q^2 \quad (1.3)$$

The equipartition theorem can be used to calculate the equilibrium amplitude:

$$\left\langle x_m \frac{\partial H}{\partial x_n} \right\rangle = \delta_{mn} k_B T \quad (1.4)$$

where H is the Hamiltonian or energy function and x_n corresponds to the thermal equilibrium for an ergodic system. A system is ergodic when the ensemble average (mean over all the possible states) is equal to the time average (mean over all time steps). Also, the equipartition theorem does not hold when the thermal energy $k_B T$ is smaller than the quantum energy spacing for a particular degree of freedom because the breakage of the energy continuum. This strong requirement does not hold for many soft matter systems. The idea behind the equipartition theorem is that the energy in thermal equilibrium is shared equally among all its degrees of freedom.

Applying equation 1.4 to 1.3, the equilibrium amplitudes of the different Fourier modes u_q^{eq} satisfy:

$$\langle u_q^{eq} \rangle = \frac{2k_B T}{L_c(\kappa q^4 + f_B q^2)} \quad (1.5)$$

δL can now be calculated as the difference between the filament's contour length and the equilibrium length.

$$\Delta L = \int dx \left[\sqrt{1 + |\partial u / \partial x|^2} - 1 \right] \simeq \frac{1}{2} \int dx |\partial u / \partial x|^2 = L_c \sum_q q^2 u_q^2 \quad (1.6)$$

Using equation 1.5 and including a factor of 2 due to both degrees of freedom of the transverse displacements from the straight conformation:

$$\langle \Delta L \rangle = k_B T \sum_q \frac{1}{\kappa q^2 + f_B} \quad (1.7)$$

The result is most conveniently expressed in terms of scaled difference between the extension at force f_B and that at zero force[2]:

$$\delta l = \langle \Delta L \rangle_{f=0} - \langle \Delta L \rangle_{f_B} = \frac{L_c^2}{\ell_p \pi^2} \sum_q \frac{\varphi}{n^2(n^2 + \varphi)} \quad (1.8)$$

where $\varphi = f_B L_c^2 / \kappa \pi^2$

Equation 1.8 can be inverted to yield a force-extension (FE) relation:[6]:

$$f(x) = \frac{k_B T}{\ell_p} \left[\frac{1}{4(1 - x/L_c)^2} - \frac{1}{4} + \frac{x}{L_c} \right] \quad (1.9)$$

where the first term in brackets dominates at the high force regime, and the last two terms $-\frac{1}{4} + \frac{x}{L_c}$ are added after the inversion of equation 1.8, to fit the linear elasticity observed experimentally at low force regime [6]. This equation diverges as $f \sim (x - L_c)^{-2}$ as the end to end distance approaches the contour length: $x \rightarrow L_c$.

This force-response, as in the case of rubber elasticity for flexible chains, is dominated by entropy at high forces. Because there are many bend configurations and only one that is perfectly straight, stretching the chain reduces its conformational entropy and thus produces an opposing force.

This force-extension curve of a single-chain is of central importance for homogeneous, and isotropic networks with affine regimes. The mechanical response of a network with these the mentioned characteristics can be derived from a single-chain force-extension curve [2].

This FE can be measured using optical tweezers or AFM. Optical tweezers are able to trap and manipulate beads of micron size with high precision[7, 8]. Biopolymer chains can be attached to these beads using linkage molecules, and then the force-extension of the chain when the force is applied can be studied by the movement of the trap in the tweezers. The extension is measured tracking the beads using microscopy.

Thanks to optical tweezers and Atomic Force Microscopy (AFM)[9] force-extension curves of many biopolymers have been measured, e.g. DNA[6], polysaccharides[10] and others. At low strain, the WLC model captures well the system behavior, but at higher forces, a Hookean extension regime must be incorporated to account for enthalpic stretching beyond the contour length. Such a model is called *extensible wormlike chain* (EWLC)[11]:

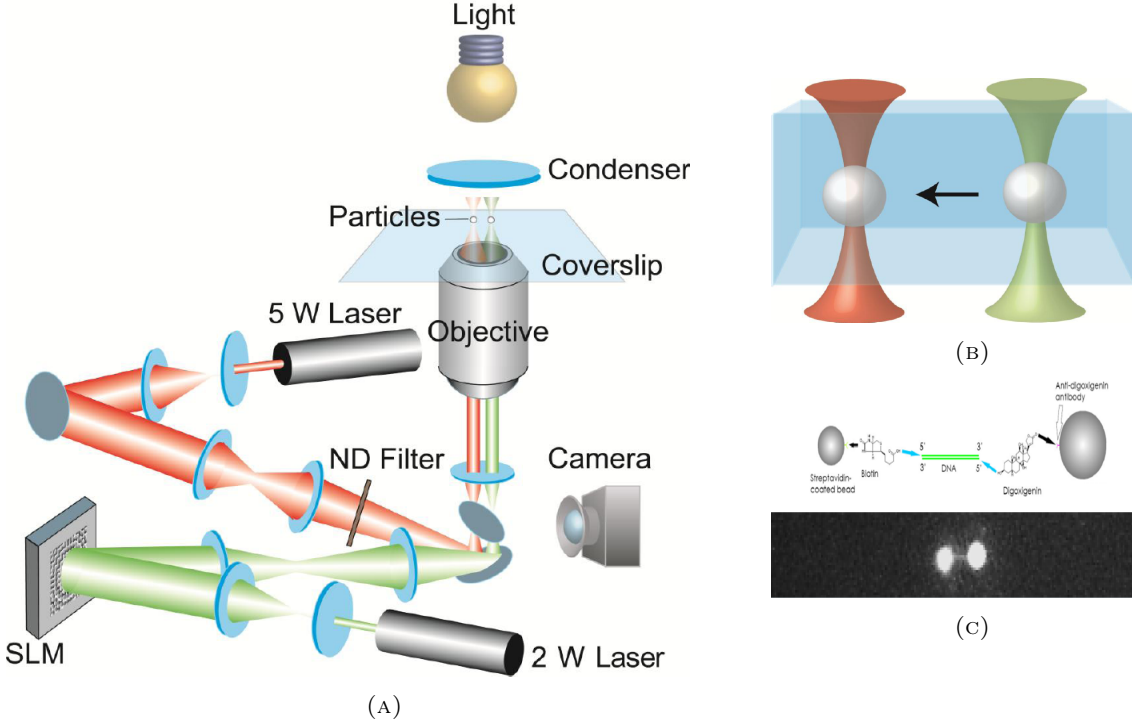


FIGURE 1.1: **a** Current optical tweezers configuration in our lab. **b** Exertion of force moving the right trap. One of the bead (red-left) is fixed by the most powerful trap. **c** Linkage of the polymer stained with fluorescence particles (double stranded DNA) to the beads, and below, visualization of the fluorescence. Images by courtesy of Sandy Suei.

$$f(x) = \frac{k_B T}{\ell_p} \left[\frac{1}{4(1 - x/L_c + f(x)/K_0)^2} - \frac{1}{4} + \frac{x}{L_c} + \frac{f(x)}{K_0} \right] \quad (1.10)$$

where K_0 is the elastic spring constant. Another model, the *clickable extensible wormlike chain* (CEWLC) incorporates extra enthalpic components of the chain, for example taking into account the conformational changes of sugar rings induced by high forces in polysaccharides[12].

1.3 Networks of semiflexible chains:

After the study of single chains, we have to study how all those chains interact with themselves, with each other and with the media where they are. The WLC is a model for ideal chains because the interaction between monomers of the chain are ignored. This is not a problem in straight (i.e. stiff) chains, but would be unrealistic in coiled chains, which are better modeled with a self avoiding walk, where distant monomers cannot occupy the same position, or with the inclusion of an excluded volume, that takes into account the interaction between monomers and the forbidden configurations.

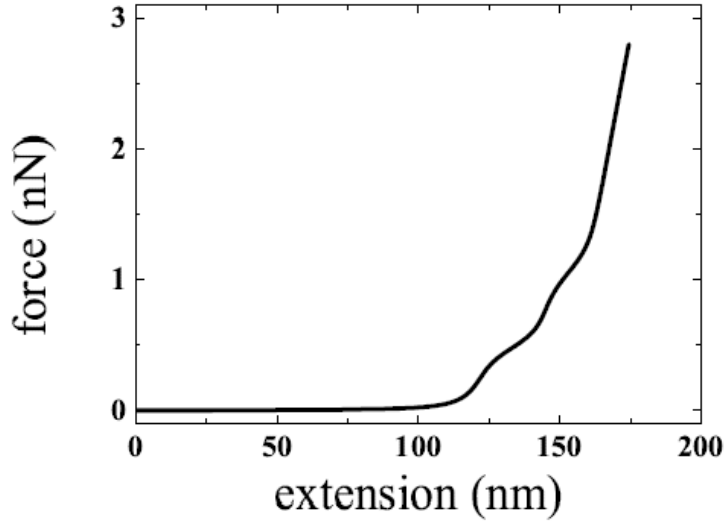


FIGURE 1.2: Force-extension curve of a CEWLC model from Ref. [4]. It fits the experimental force-extension curve of a single polysaccharide (pectin) chain [12]

The way the monomers of the chain interact with other monomers or with the solvent has been studied for long time. The Flory interaction parameter χ which depends on temperature, and pressure of the system, is a way to characterize it.

$$\chi = \chi_{MS} - \frac{1}{2}(\chi_{MM} + \chi_{SS}) \quad (1.11)$$

where χ_{MM} is related with monomer-monomer interactions. χ_{MS} with monomer-solvent, and χ_{SS} with solvent-solvent interactions.

The relation between solvent, and monomers interaction can be classified using χ :

1. Athermal. $\chi = 0$. Solvent is very similar to other monomers.
2. Good solvent. $\chi < 1/2$. Monomers of the chain prefer to interact with the solvent than with other monomers.
3. Poor solvent. $\chi > 1/2$. Monomers prefer to be closer to other monomers. For example, hydrophobic materials in water solvent.
4. Theta solvent. $\chi = 1/2$. Occurs at a temperature $T = \Theta$, and corresponds to the exact cancellation between steric repulsion and van der Waals attraction between monomers.

The excluded volume:

$$v = (1 - 2\chi)a^3 \quad (1.12)$$

where a is the effective length between monomers, is zero in theta solvents, and chains behave as nearly ideal.[\[13\]](#)

Also, the concentration of polymers has important effects on the system. The overlap threshold parameter c^* acts as an order parameter to distinguish dilute polymer solutions, where the coils are separate, and more concentrated solution where the chains overlap. This threshold is not sharp, it is more properly defined as a crossover between regions, but the scaling properties of c^* with concentration are essential.

1. Dilute. $c < c^*$. Chains can be studied as an ideal gas, with almost no interaction between them.
2. Dense. $c > c^*$.
3. Semi-dilute. The chains do overlap but the polymer fraction $\phi = ca^3$ is still low $\phi^* \ll \phi \ll 1$. Since ϕ is small, the monomer-monomer interaction can be described very simply, such as with the excluded volume parameter v in [1.12](#). In the case of dense systems, more complex relations used in liquids/fluids systems are needed.

In dilute systems, polymer chains flow through the solvent acting similar to a liquid state, in the other hand, dense system are more like solids, but they are not completely frozen. The transition between both regimes is called the gelation point. Th gelation is a connectivity transition, where parts of the system becomes correlated by a path of inter-connexion. In the gelation point, the chains stop flowing as a liquid. It can be thought as an incipient gel, where a cluster of chains started to form (giant cluster), connecting the boundaries of the material. This incipient gel will behave differently to a fully connected network, where the giant cluster is connecting all the chains.

It is not easy to give a closed definition of gels[\[3, 13\]](#), one of them that is particularly appealing reads: the point in the phase space of parameters, where a significant correlation, between two points in the spatial domain appears over a length scale of the system size.

1.4 Bulk Rheology

1.4.1 Linear viscoelasticity

Any elastic material can be studied as a linear spring, whereby the extension (strain) γ is proportional to the applied stress σ (force/area) : $\sigma = G'\gamma$, where G' is called the elastic modulus, and it is a measure of sample's elasticity.

However, the material response to the force is not always that simple. Some materials such as silk, rubber, and glass are subjected to a shear stress and the corresponding Hooke's law deformation occurs, however it is followed by an unexpected continuous deformation termed as "creep". Upon removal of the shear stress, certain materials come back to the initial state, while other are permanently deformed. The phenomena of a time dependant shear response is called viscoelasticity. [14].

One way to study the deformation and flow of viscoelastic materials under strain, is to externally apply a small amplitude sinusoidal strain: $\gamma(t) = \gamma_0 \sin(\omega t)$. It can be expected (not always, e.g. non-linear regime) that the stress output of the material will be:

$$\sigma(t) = \sigma_0 \sin(\omega t + \delta) \quad (1.13)$$

This stress output can be analyzed by decomposition into two waves of frequency ω , one in phase with the strain, and the other $\pi/2$ out of phase:

$$\sigma(t) = \sigma'(t) \sin(\omega t) + \sigma''(t) \cos(\omega t) \quad (1.14)$$

where $\tan \delta = \sigma''(t)/\sigma'(t)$. This allows the definition of two moduli: $G' = \sigma'(t)/\gamma_0$ and $G'' = \sigma''(t)/\gamma_0$ which are the elastic (i.e. storage) modulus and viscous (i.e. loss) modulus respectively.

The stress can be expressed in terms of these two moduli:

$$\sigma(t) = G' \gamma_0 \sin(\omega t) + G'' \gamma_0 \cos(\omega t) = G' \gamma(t) + \frac{G''}{\omega} \dot{\gamma}(t) \quad (1.15)$$

G'' has a physical meaning, it is associated with the energy dissipation per cycle of deformation, which provides an indication of viscous loss [14]. Note also that $\eta' = \frac{G''}{\omega}$ corresponds to the dynamic viscosity, used to characterize stress-response in pure liquids.

1.4.2 Nonlinear behaviour: strain stiffening

At small strain the stress-response is linear, but when the strain amplitude is increased beyond a certain point, non-linear responses arise. In this situation, the premise imposed in Eq.1.13 that the stress response is linear does not hold, however the qualitative study of G' and G'' still provides information about the nature of the non-linearities. [2, 15–18].

Fig. 1.3 shows that most of the biopolymer networks share the same non-linear behaviour at high strains: strain-stiffening. In plain words, the higher the strain is, the tougher the material becomes, or the more you pull, the harder it is to pull more. The universal behavior in biopolymers may have an evolutionary origin, due to the role of the strain-stiffening in biological functions such as cell motility and mechano-transduction.

One way to study strain-stiffening experimentally is to use a *large amplitude oscillatory shear rheology* (LAOS) [19]. This technique consists in pre-stress the network at a selected constant strain, and then, study the frequency response applying other small amplitude sinusoidal strain, as in the linear viscoelasticity situation.

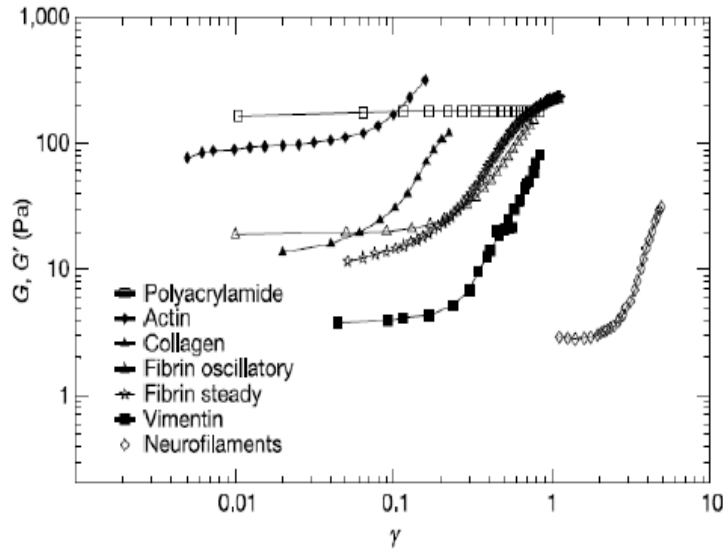


FIGURE 1.3: Elastic shear moduli versus strain from Ref.[2]. Strain-stiffening behavior in different cross-linked biopolymer networks [15].

The research on strain-stiffening in biopolymers is active [2, 18, 20–24]. Stiffening can result from the response of the chain between cross-links, from alterations in the network structure, or from both.

In a seminal paper, Storm et al. [2] showed that: for low to middle strains, under the assumptions that networks are homogeneous, isotropic, and strain uniformly (affine deformations), the strain-stiffening is primarily due to the longitudinal stiffening of the single chains (i.e. the non-linearity of the force response curve such as Fig.1.2).

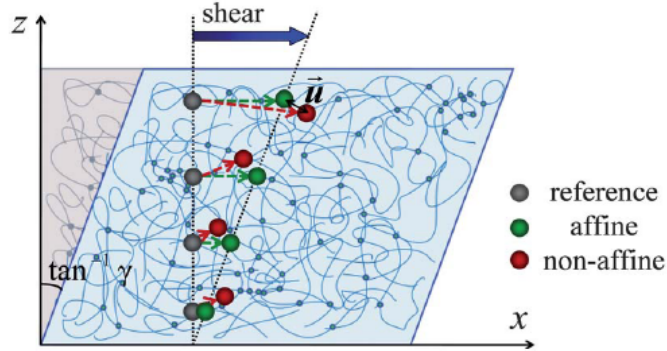


FIGURE 1.4: Difference between affine and non-affine deformation of a network under shear.
Figure adopted from references [25, 26]

The affine deformation assumption (Fig.1.4) is well-known in network models for rubber elasticity, and allows for a relatively simple description of the overall network response on the basis of the behavior of a single filament. Predicting the viscoelastic response of networks under the affine assumption can be considered straightforward from current literature [27]. But, this assumption does not hold for many systems.

The small-strain affine deformation assumption in two-dimensional networks of straight filaments has been studied in great detail by Head, Levine, and MacKintosh [20, 28] and Wilhelm and Frey [24], (Fig.1.5) who conclude that network deformation is non-affine for compliant, low-density networks and affine for stiff, high-density networks.

Also, Chandran and Barocas [29] showed that non-affinity governs individual fibril kinematics when the load transfer is from fibril to fibril, with no matrix of fibrils involved.

But non-affinity is challenging to model, and it is the goal of this thesis to provide a method to extract network architecture from real images, to be used as a starting scaffold for the many approaches to simulate dynamics of non-affine regimes, and dynamics with more actors, such as cross-linking molecules.

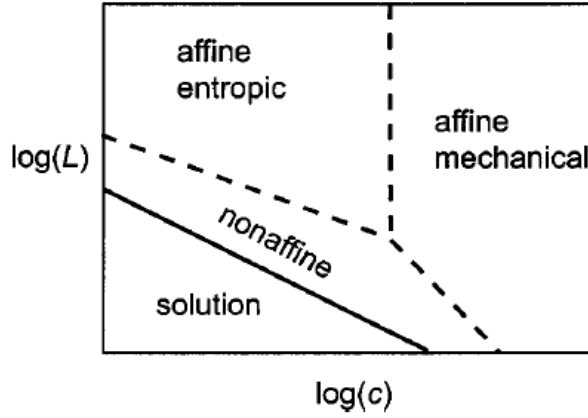


FIGURE 1.5: Figure from Head, Levine, and MacKintosh [28]. Diagram showing the various elastic regimes in terms of molecular weight (L) and concentration (c). The solid line represents the rigidity percolation transition, where rigidity first appears at macroscopic level.

1.5 Network Structure

The information needed to fully describe a network structure is [30]:

1. Distribution in the distance between junctions/nodes, i.e. end-to-end distances.
2. Distribution in the fully extended length of fibers/chains/edges, i.e. contour lengths.
3. Network connectivity, i.e. degree of nodes, representing the number of neighbors or the number of chains attached to a junction.
4. Orientation distribution, i.e. angles between adjacent edges.

I will go through the most used methods to emulate network structures used in dynamic simulations that explore strain and stress responses, pointing the simplifications made, and the associated limitations.

1.5.1 Mikado Networks

Ubiquitous in the recent past but still present, is to start the model with a random and homogeneous initialization of the network, such as Mikado models [24, 31–35], where fibers of fixed length are thrown into a simulation box until a certain density of junctions/intersections is reached (see Figure 1.6). Other models use similar random generation, but in 3D [36].

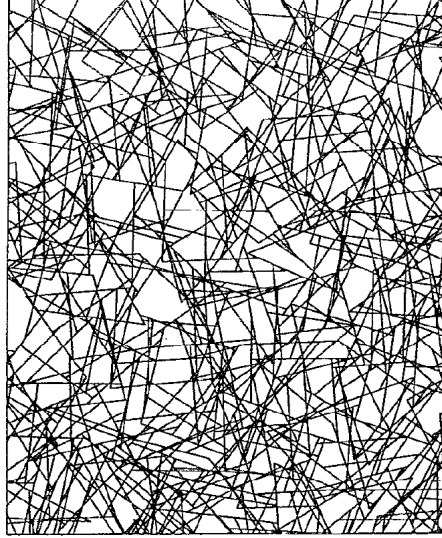


FIGURE 1.6: Mikado network used as a starting scaffold for dynamics simulations, from Ref. [31].

The distribution of end-to-end distances between junctions can be controlled adding more fibers, but the homogeneous nature of the process inhibits long connections.

The contour length is imposed after the junctions are created, adding arbitrary bending points in the existing chains.

The details of network connectivity are ignored, relying on a mean connectivity between 3 and 4 that is generated in the construction [34].

The angle distribution is derived from the process, but no extra relation with real networks is provided. Pre-stressing the network can be used to change angles.

1.5.2 Lattice Models

Lattice models with different geometries, such as triangular and honey-comb in 2D, and face-centered-cubic (fcc) in 3D have been used [34, 37–39]. The unrealistic connectivity imposed by the fcc lattice in 3D (see Figure 1.7) is trimmed down to experimentally found values by randomly deleting branches.

The end-to-end distances and the contour lengths can be fully controlled, but the connectivity is compromised, the distribution might be too narrow for simple lattices, (honey-comb, triangular), or rely on random dilutions for over-connected lattices (fcc) [34].

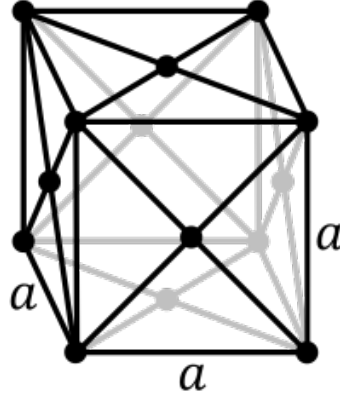


FIGURE 1.7: Unit cell of a face-centered-cubic lattice, from Ref. [40]

1.5.3 Eight-chain Models

The eight-chain network model [30, 41] with a extremely simplified network connectivity is able to successfully capture the mechanical properties of non-affine regimes under shear. However, it is unable to capture other observed behaviours like negative normal stresses [42], or to model networks with cross-link dynamics, and other micro-network events, which are out of its scope. The eight, or even three-chain models, use non-linear single-chain force-extension curve to model each of its chains, see Figure 1.2. When sheared, the chains can rotate (see Figure 1.8).

In terms of the network structure, this model is surprisingly effective given its simplicity. The main reason for its success is its ability to accommodate rotation response under shear, instead of only stretching of fibers that would be the only response in the affine regime.

The eight-chain model coupled with single chain non-linear force-extension curves shows a good example of how network architecture can have an impact in the mechanical response. Similar simulations, but carried out with a fully detailed network architecture extracted from real samples would be interesting. The network structure of these models are currently over-simplified:

The distribution of end-to-end distances is a delta function, taking a mean distance for each chain and same is true for the contour length. The connectivity is eight, or three for the similar three-chain case. The angle between chains is fixed as well, and can be re-oriented with pre-stress.

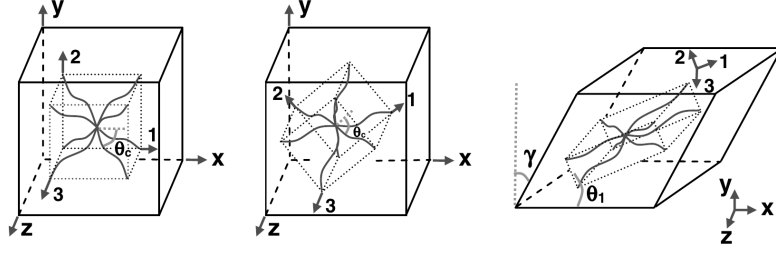


FIGURE 1.8: Eight-chain model under shear, from left to right: initial state, a rotation, and a simple shear, from Ref. Palmer and Boyce [30]

1.6 Motivation

Non-affinity and network architecture are closely related, which is the **main motivation** behind this work: establish analytical methods to study images with a view to gathering the geometry of networks experimentally. Such geometries could then be used directly in dynamic simulations methods, instead of relying on over-simplified network generators.

The well-established and trusted method to gather structural information about networks is scattering, particularly small-angle x-ray scattering (SAXS). Extracting information such as porous size and persistence length are possible with these techniques. However, because of its averaging nature, it is impossible to gather more detailed information about the specific geometry and connectivity of the network.

I will rely on gathering structural information directly from images taken from a range microscopy techniques depending on the size of the biopolymer under study. One of the concerns of this approach, shared specially among the scattering community is how trustable the images are, i.e. if they provide a true representation of the material taken into account that the sample has been modified for the requirements of microscopy preparation. This will be answered, providing a way to compare SAXS data with images, setting the scale at which both techniques give similar results and providing a size range where image features can be trusted.

But before, and in order to study network geometry reliably, an image analysis tool-set will be provided with algorithms that can be applied to volumetric data, such as tomography and stacks of images. State-of-the-art wavelet analysis algorithms will be extended into 3D using a high-performance computing language, which will aid in our task of gathering the network architecture from images. Hence realistic in-silico networks will be generated informed by real 3D images. These in-silico networks with full details, can then be used by theoreticians

as starting points for non-affinity modeling, instead of the currently used over-simplified architectures. We finish by providing a method to reconstruct these networks fully in-silico, just with statistical distributions gathered from the networks extracted from real images. This will allow the exploration of optimal network architectures for different processes to be carried out in-silico, but rooted on networks found in nature.

The research question that this work aim to answer is:

- How realistic network architectures can be reliably extracted from 3D images?

The main motivation is to go beyond simplistic models for theoretical approaches, using instead real geometries extracted from images.

2. Isotropic and Steerable Wavelets in N Dimensions. A multiresolution analysis framework

2.1 Motivation

In our quest to extract structural information of biopolymer networks from microscopy images we need to pre-process these images to get rid of intrinsic noise and spurious information that arises from the different microscope techniques. One of the sharpest tools in this area is wavelet analysis, but most of the software in this field is only available in scripting languages, (especially the commercial Matlab) so it is not available for high-performance applications or to export to wider communities. Also, it is usually only implemented for 2D images. In this chapter we describe the implementation of isotropic wavelets for the `c++` image analysis library Insight ToolKit (ITK), which is open source, the most used library for medical image analysis and the best tool to handle volumetric images, providing mechanisms and helpful abstractions that allow to develop algorithms and apply them to 2D **and 3D** images.

We will use the wavelet phase analysis introduced here as a denoiser for 2D images in the next chapter and later on in the thesis, it will aid in the binarization of 3D images in preparation for the extraction of graph parameters.

2.2 Abstract

This document describes the implementation of the external module `ITKIsotropicWavelets`, a multiresolution (MRA) analysis framework using isotropic and steerable wavelets in the

frequency domain. This framework provides the backbone for state of the art filters for denoising, feature detection or phase analysis in N-dimensions. It focuses on reusability, and highly decoupled modules for easy extension and implementation of new filters, and it contains a filter for multiresolution phase analysis.

The backbone of the multi-scale analysis is provided by an isotropic band-limited wavelet pyramid, and the detection of directional features is provided by coupling the pyramid with a generalized Riesz transform. The generalized Riesz transform of order N behaves like a smoothed version of the N th order derivatives of the signal. Also, it is steerable: its components impulse responses can be rotated to any spatial orientation, reducing computation time when detecting directional features.

This chapter is accompanied with the source code, input data, parameters and output data that the author used for validating the algorithm described in this paper. This adheres to the fundamental principle that scientific publications must facilitate reproducibility of the reported results.

2.3 Wavelet Multiresolution Analysis

2.3.1 Introduction

This module is a generalization to N dimension for ITK of the work made by: [43–49].

Learning about wavelets from research papers is not easy. The topic is rich and deep, the same tool has spanned different research fields, from theoretical physics, to seismology and signal processing. Yves Meyer has won the Gauss Price (2010), and Abel Price (2017) in Mathematics for “his pivotal role in the development of wavelets and multi-resolution analysis”. The detection of Gravitational waves with the LIGO experiment involved the use of wavelets to analyse the signals.

Stephan Mallat [50] had also a fundamental role in developing the Multiresolution Analysis (MRA) and making the implementation available for applications.

The steerable framework [43, 44] is widely used in applications to rotate the filter bank to any direction, avoiding expensive computations.

We will also implement more recent work [46–48] that uses the Riesz transform (a natural generalization of the Hilbert transform for N dimensions) to provide a generalization and extra flexibility to the steerable pyramid.

2.3.2 Motivation: spatial and frequency resolution

Wavelets share the same motivation as the short-time FFT, or the windowed Fourier transform: get a representation of the signal/image/function that is well localized in the space **and** frequency domains. The Heisenberg Uncertainty principle applies here (where it is called the Heisenberg-Gabor limit: $\Delta t \cdot \Delta f \geq \frac{1}{4\pi}$) limiting the simultaneous resolution of a signal in time-frequency domains. The Fourier transform, which is the representation of the signal in the basis of harmonic functions $\{\sin(f), \cos(f)\} \forall f \in \mathbb{R}$, is completely localized in frequency $\Delta f = 0$, but has global support in space, i.e infinitely de-localized in space $\Delta t = \infty$.

A few examples to illustrate the concept: In a discrete case, such as an image, this means that the FFT has the highest resolution on frequency, but where, in the spatial domain, a specific frequency is only known to occur ‘somewhere’, bounded by the size of the image (Figure 2.1, 2.2a). Or in signals occurring only in one time-lapse: they can’t be differentiated from the same signal occurring continuously, so they will share similar spectra representation (Figure 2.1). In the same line, non-stationary signals which frequency depends on time will give a spectrum where all those frequencies are represented, but there is no information about when, in time, the change of frequency occurred.

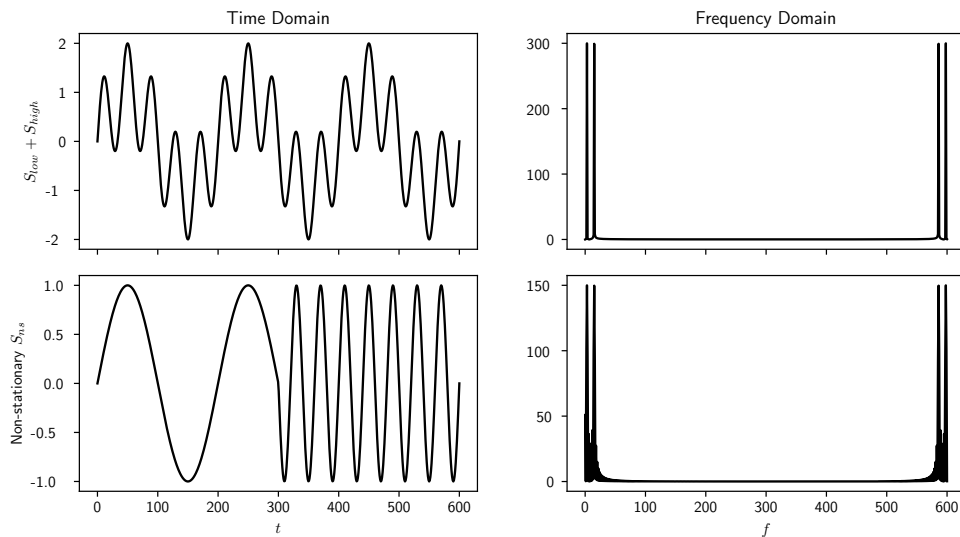


FIGURE 2.1: The changes over time/space of a signal are not captured by the FFT.

The short-time FFT adds spatial/time localization dividing the original signal in small, consecutive segments, and applying the FFT to each of those. The problem with this approach, besides being computationally expensive, is that events shorter than the time window are still not resolved and that the width of the segment is constant (Figure 2.2b). Would it be possible to have better spatial resolution for some components of the frequency spectra, such as high frequency components (Figure 2.2c)?

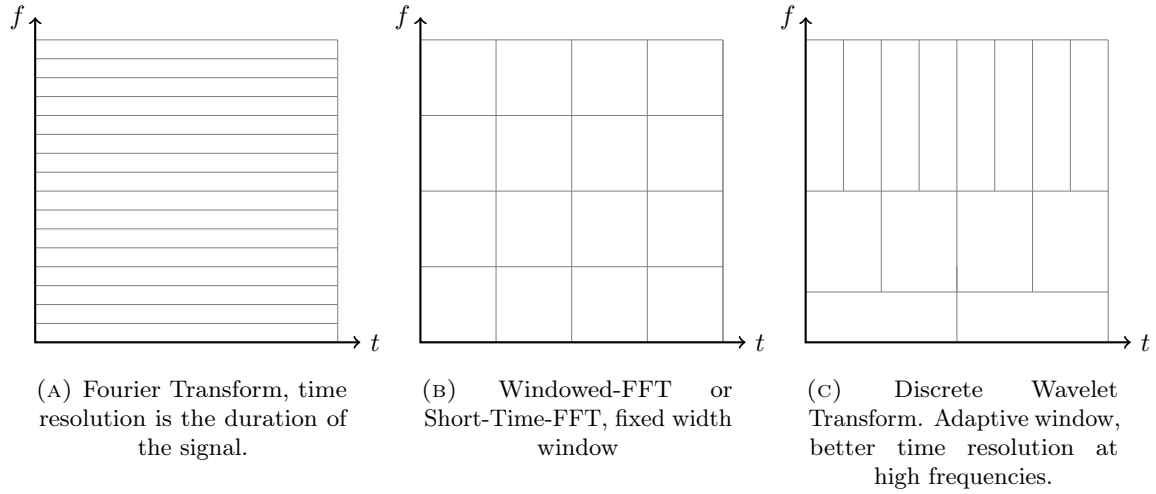


FIGURE 2.2: Simultaneous spatial and frequency resolution of different transformations. $\Delta t, \Delta f$

2.3.3 Wavelet transformation

Consider the following wavelet decomposition $\forall f \in L_2(\mathbb{R}^d)$

$$f(\mathbf{x}) = \sum_{s \in \mathbb{Z}} \sum_{\mathbf{l} \in \mathbb{Z}^d} \langle f, \psi_{s,\mathbf{l}} \rangle \psi_{s,\mathbf{l}}^*(\mathbf{x}) \quad (2.1)$$

The family of functions $\{\psi_{s,\mathbf{l}}\}$ is a wavelet *frame* (Def. 2.2), constructed by means of *translations* and *dilations* (Def. 2.1) of the *mother wavelet* function ψ .

$$\psi_{s,\mathbf{l}}(\mathbf{x}) = |\det(A)|^{-s/2} \psi(A^s \mathbf{x} - \mathbf{l}) \quad (2.2)$$

Each dilation, defined by the dilation matrix A , squeezes or stretches the mother wavelet, acting as a change of scale. The translation operator moves and centers the location of the mother wavelet ψ . If the dilation matrix is diagonal with the same dilation factor a in all

dimensions, $A = a\mathcal{I}_d$, then Equation 2.2 becomes:

$$\psi_{s,\mathbf{l}}(\mathbf{x}) = a^{-d \cdot s/2} \psi(a^s \mathbf{x} - \mathbf{l}) \quad (2.3)$$

The mother wavelet function has to have finite energy $\psi \in L_2(\mathbb{R}^d)$, i.e. $\int_{-\infty}^{\infty} |\psi(\mathbf{x})|^2 d\mathbf{x} < \infty$.

Definition 2.1 (Dilations and Translations). Given a function $f \in \mathbb{R}$ we define the dilation and translation operators[51]:

$$\text{Dilation:} \quad D_a f(x) := |a|^{-1/2} f(x/a) \quad \text{for } a \in \mathbb{R} \setminus \{0\}$$

$$\text{Translation:} \quad T_b f(x) := f(x - b) \quad \text{for } b \in \mathbb{R}$$

\mathbb{R}^d generalization:

Given $f \in L_2(\mathbb{R}^d)$, $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $\mathbf{x} \in \mathbb{R}^d$, the dilation scalar a is replaced by a dilation matrix $A = a\mathcal{I}_d$ [52]. The dilation matrix A is expansive, having all its eigenvalues $|\lambda_i| > 1$, so it is invertible. For the translation operator, the scalar b is replaced for the vector \mathbf{b} .

$$\text{Dilation:} \quad D_A f(\mathbf{x}) := |\det(A)|^{-1/2} f(A\mathbf{x}) \quad A \text{ expansive matrix}$$

$$\text{Translation:} \quad T_{\mathbf{b}} f(\mathbf{x}) := f(\mathbf{x} - \mathbf{b}) \quad \text{for } \mathbf{b} \in \mathbb{R}^d$$

Definition 2.2 (Frame). A family of functions $\{\phi_{\mathbf{k}}\}_{\mathbf{k} \in \mathbb{Z}^d}$ is a *frame* of $L_2(\mathbb{R}^d)$ if and only if there exists two positive constants $A, B < \infty$ such that:

$$A\|f\|^2 \leq \sum_{\mathbf{k} \in \mathbb{Z}^d} |\langle \phi_{\mathbf{k}}, f \rangle|^2 \leq B\|f\|^2, \forall f \in L_2(\mathbb{R}^d) \quad (2.4)$$

The frame is *tight* if $A = B$. If $A = B = 1$ we have a *Parseval frame* that satisfies the decomposition/reconstruction formula:

$$f = \sum_{\mathbf{k} \in \mathbb{Z}^d} \langle \phi_{\mathbf{k}}, f \rangle \cdot \phi_{\mathbf{k}}, \forall f \in L_2(\mathbb{R}^d) \quad (2.5)$$

which has the same form as the expansion using an orthonormal basis, however in the frame generalization, the family $\phi_{\mathbf{k}}$ may be redundant.

2.3.4 Wavelet Pyramid

A band-limited pyramid is created by applying, at each level, a low-pass filter h_0 and downsampling by a scale factor of two, see Figure 2.3a. Because the wavelet frames are tight, we can get perfect reconstruction applying the inverse pyramid from Figure 2.3b. In Figure 2.3 we use two levels and two high pass subbands h_1, h_2 . The results of this pyramid are the detail coefficients $d_{s,h}$, where $s \in \{1, \dots, \text{Levels}\}$, and $h \in \{1, \dots, \text{HighPassSubBands}\}$. Note that these details are in the frequency domain, if the original image was given in spatial domain, an inverse Fourier transform for each $d_{s,h}$ must be performed to get the wavelet coefficients in the spatial domain.

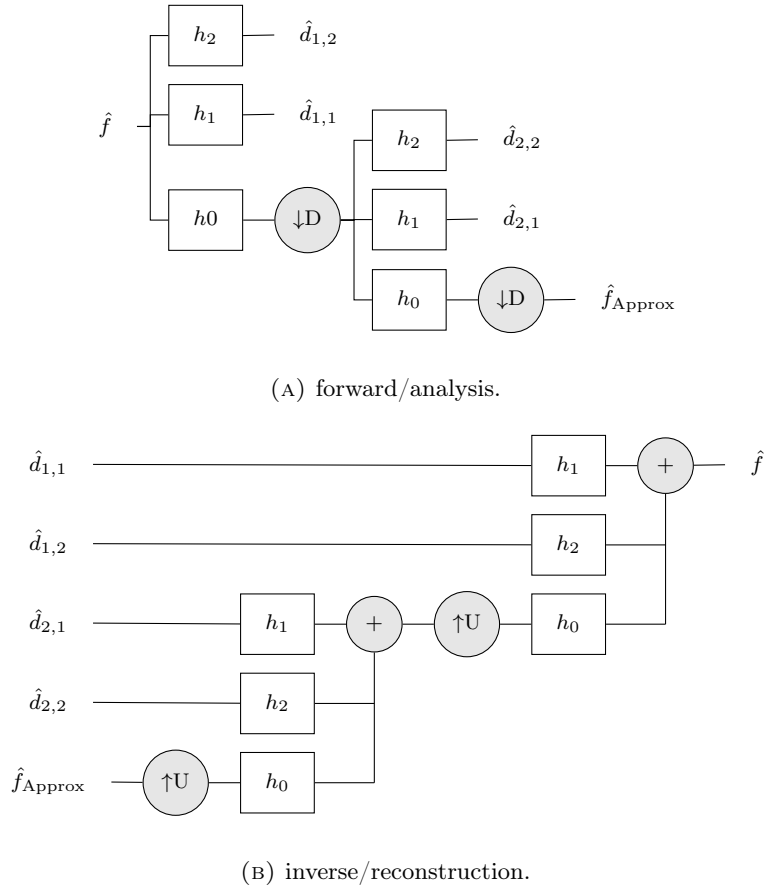


FIGURE 2.3: Forward 2.3a and Inverse wavelet 2.3b two-level pyramid with two high pass sub-bands.

Usually in the literature the only one high-pass sub-band is used, the wavelet filter bank consists then in one high pass filter and one low pass filter (see Figure 2.4). The advantages of the subbands is that they provide more frequency resolution [46], at expenses of more

computation time, and also they might generate spatial domain distortions due to multiple sharp cutoffs. For phase detection (see the application in Figure 2.9), they are fundamental.

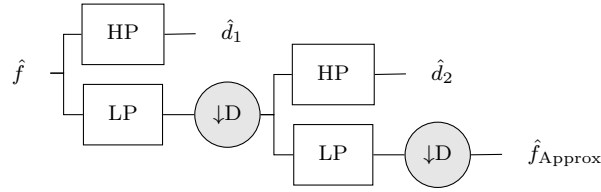


FIGURE 2.4: Forward wavelet pyramid with a classic two level filter bank and only one sub-band, HP is the high pass filter, and LP low pass

2.3.5 Isotropic wavelets

These wavelets are non-separable in Cartesian coordinates, depending on $\|\omega\|$. Isotropic wavelets have the same mother wavelet for each scale. All of the four implemented wavelets fulfill the conditions of Prop 2.3, their shapes differ in the spatial decay, and vanishing moments. Which one is better depends on the application, and developing new wavelets is a research topic. The advantage of isotropic wavelets is that they are steerable, but in order to get directionality features, the wavelets are coupled with other filters that gather directional information, such as the Riesz transform, see section 2.4.

The present requirement that the mother wavelet is isotropic constraints the wavelet-design. The goal is to generate a tight wavelet frame and have perfect reconstruction with the inverse pyramid.

Proposition 2.3. *Conditions for the mother wavelet ψ to generate a tight wavelet frame [46–48]:*

Let $h(\omega)$ be a radial frequency profile such that:

1. (Band-limitedness): $h(\omega) = 0, \forall \omega > \pi$.
2. (Riesz Partition of Unity): $\sum_{i \in \mathbb{Z}} \left| h\left(\left((A^T)^{-1}\right)^i \omega\right) \right|^2 = 1$.
If $A = a\mathcal{I}_d$: $\sum_{i \in \mathbb{Z}} |h(a^i \omega)|^2 = 1$. [46, 53]
3. (Vanishing Moments): $\left. \frac{d^n h(\omega)}{d\omega^n} \right|_{\omega=0} = 0$, for $n = 0, \dots, N$.

If the mother wavelet ψ is defined by $\hat{\psi}(\omega) = h(\|\omega\|)$, then it generates a tight wavelet frame in $L_2(\mathbb{R}^d)$, where $\hat{\psi}$ is the ND Fourier transform of ψ .

Condition 2 guarantees that the tiling from all the scales fills the frequency domain, see Figure 2.5.

VOW, variance-optimal wavelets [49] :

$$h(\omega) = \begin{cases} \sqrt{\frac{1}{2} + \frac{\tan(\kappa(1 + 2\log_2 \frac{2\omega}{\pi}))}{2\tan(\kappa)}}, & \omega \in [\frac{\pi}{4}, \frac{\pi}{2}[\\ \sqrt{\frac{1}{2} - \frac{\tan(\kappa(1 + 2\log_2 \frac{\omega}{\pi}))}{2\tan(\kappa)}}, & \omega \in [\frac{\pi}{2}, \pi] \\ 0, & \text{otherwise} \end{cases}$$

where $\kappa \in [0, \frac{\pi}{2}]$ is found to be 0.75

Held [46] :

$$h(\omega) = \begin{cases} \cos\left(2\pi q_n\left(\frac{\omega}{2\pi}\right)\right), & \omega \in]\frac{\pi}{4}, \frac{\pi}{2}] \\ \sin\left(2\pi q_n\left(\frac{\omega}{4\pi}\right)\right), & \omega \in]\frac{\pi}{2}, \pi] \\ 0, & \text{otherwise} \end{cases}$$

where q_n is a polynomial function of order n

Simoncelli [44, 54] :

$$h(\omega) = \begin{cases} \cos\left(\frac{\pi}{2} \log_2 \frac{2\omega}{\pi}\right), & \omega \in]\frac{\pi}{4}, \frac{\pi}{2}] \\ 0, & \text{otherwise} \end{cases}$$

Shannon:

$$h(\omega) = \begin{cases} 1, & \omega \in [\frac{\pi}{2}, \pi] \\ 0, & \text{otherwise} \end{cases}$$

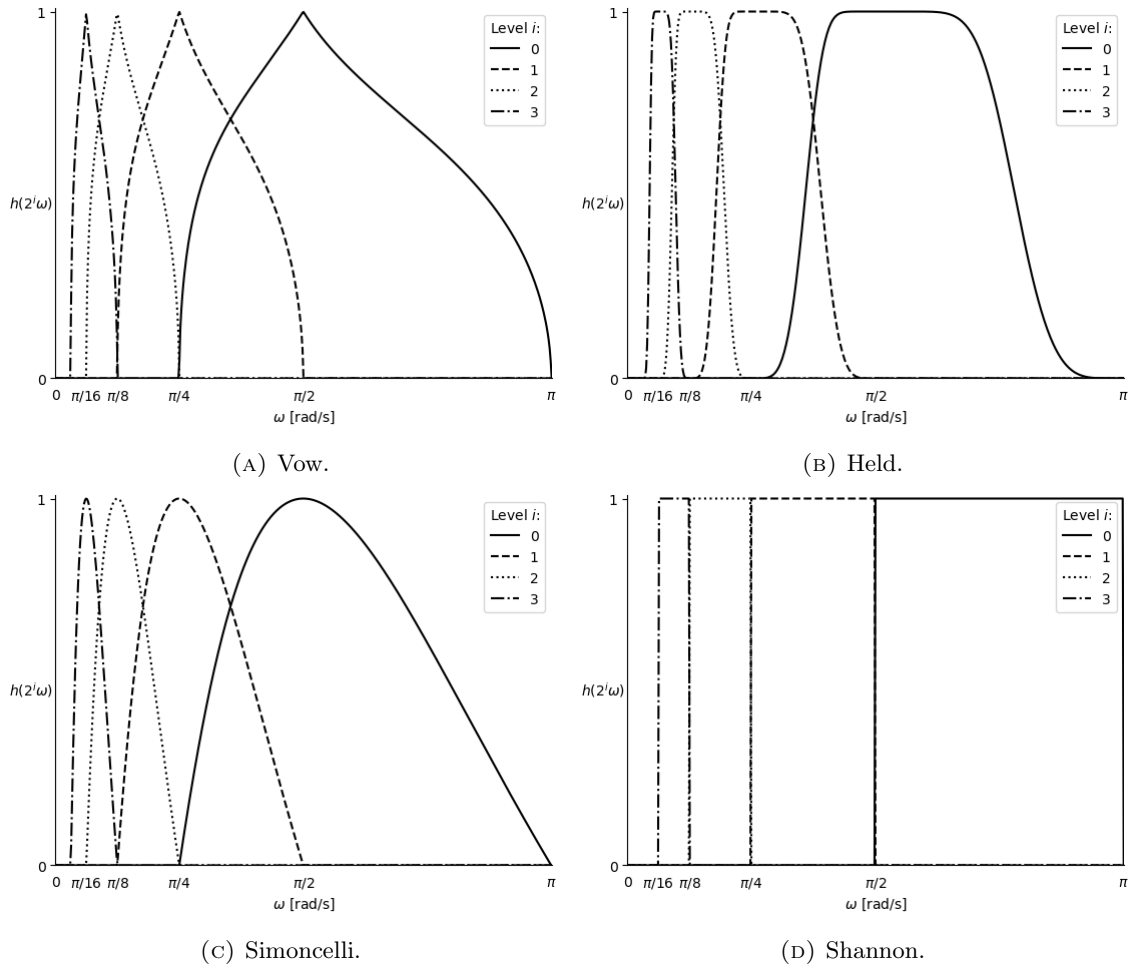


FIGURE 2.5: Tiling of the frequency domain by isotropic wavelets when the dilation factor is 2. All the wavelets fulfill the conditions from Proposition 2.3. The mother wavelets are represented at $i = 0$.

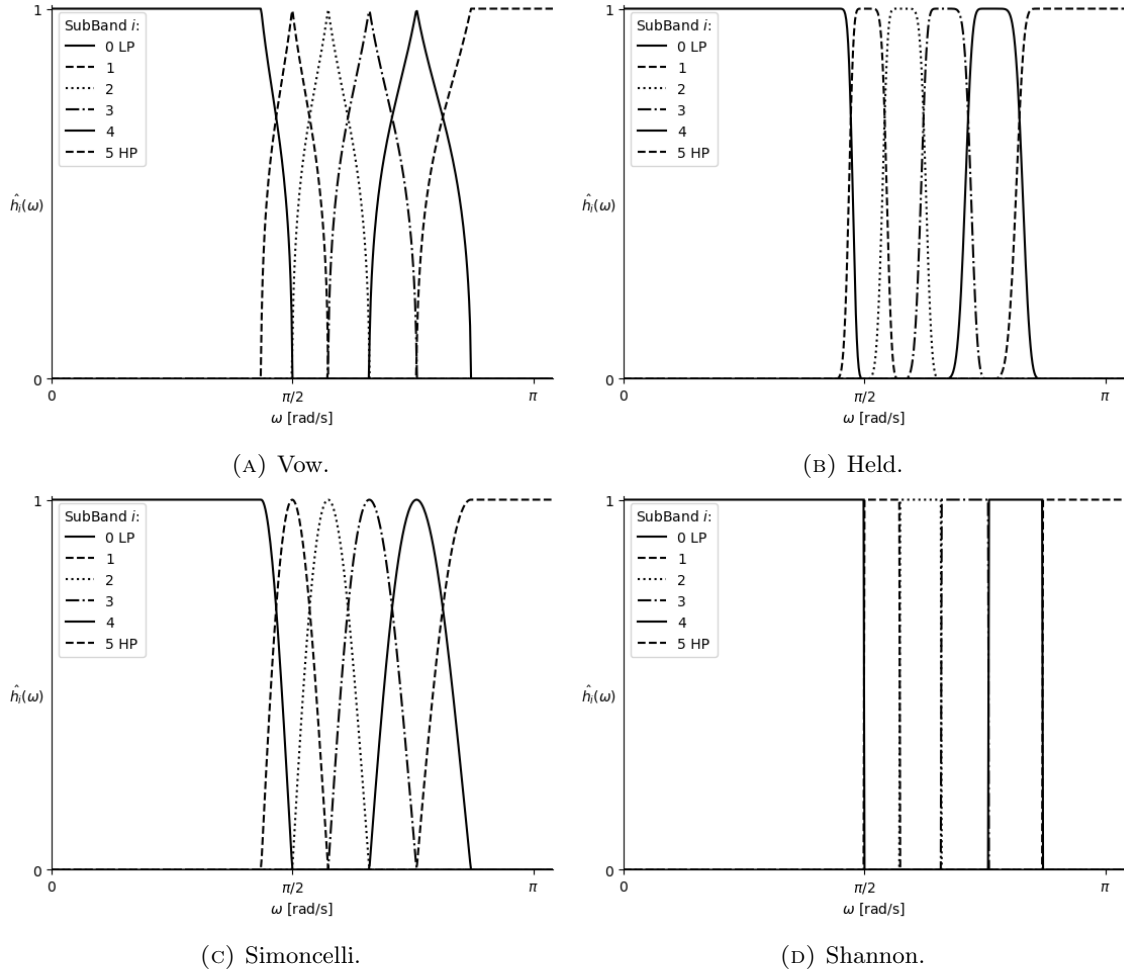


FIGURE 2.6: Shape of SubBands when $\text{HighPassSubBands} = 5$, sub-bands increase the frequency resolution, but can generate extra artifacts in the spatial domain due to the sharp frequency cut-offs.

2.4 Riesz Transform

The forward or analysis wavelet pyramid outputs a set of wavelet coefficients with information about each scale. A steerable filter [44, 46, 48] can be applied to the output of the wavelet pyramid. This is the main purpose of the isotropy of the mother wavelet, the steerable filter is used to select the orientation where the feature of interest is maximum.

There are mathematical constraints in the steerable filters that can be coupled with the wavelet pyramid. Read more: [48, 55].

One of the transform that can be coupled to the pyramid is a Riesz transform \mathcal{R} of order $N = 1$

$$\mathcal{R}f(\mathbf{x}) = \begin{pmatrix} \mathcal{R}_1 f(\mathbf{x}) \\ \vdots \\ \mathcal{R}_d f(\mathbf{x}) \end{pmatrix} \xleftrightarrow[\mathcal{F}^{-1}]{\mathcal{F}} \hat{\mathcal{R}}\hat{f}(\boldsymbol{\omega}) = -j \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \hat{f}(\boldsymbol{\omega}) \quad (2.6)$$

where d is the image dimension and the number of components of the Riesz transform of order $N = 1$. $j = \sqrt{-1}$ and $\mathcal{F}, \mathcal{F}^{-1}$ are the forward and inverse Fourier Transform.

The Riesz transform is the generalization of the Hilbert transform for ND. The Hilbert transform \mathcal{H} is used to generate the *analytic signal* in 1D.

$$f_a(x) = f(x) + j\mathcal{H}f(x) \\ \mathcal{H}f(x) = (h * f)(x) \quad \xleftrightarrow[\mathcal{F}^{-1}]{\mathcal{F}} \quad \hat{\mathcal{H}}\hat{f}(\omega) = -j \operatorname{sgn}(\omega) \hat{f}(\omega) = -j \frac{\omega}{|\omega|} \hat{f}(\omega) \quad (2.7)$$

2.4.1 Monogenic Signal

There is no direct generalization of the analytical signal in ND, but we can generate a signal with similar properties, called the **Monogenic signal** using the Riesz transform of order 1 [45, 56]. The Monogenic signal is a set formed by the original signal and the D-components of the first order Riesz transform.

$$f_a = \{f, \mathcal{R}_x, \dots, \mathcal{R}_d\} \quad (2.8)$$

The amplitude A and the phase P at each location of the Monogenic signal are:

$$A(\mathbf{x}_0) = \sqrt{f(\mathbf{x}_0)^2 + A_R(\mathbf{x}_0)^2} \quad (2.9)$$

where $A_R(\mathbf{x}_0) = \sqrt{\sum_{i=1}^N \mathcal{R}_i(\mathbf{x}_0)^2}$

$$P(\mathbf{x}_0) = \operatorname{atan2}(A_R(\mathbf{x}_0), A(\mathbf{x}_0)) \quad (2.10)$$

The Monogenic signal can be used to perform local phase analysis for feature detection in ND.

2.4.2 Generalized Riesz Transform

The Riesz transform will map any frame of $L_2(\mathbb{R}^d)$ into another one [46, 57]. This mapping property allows to couple the Riesz transform of any order with the wavelet pyramid and if the wavelet function is isotropic, to get perfect reconstruction when performing the inverse transform. The Riesz transform of order N generates a vector containing smoothed directional N th derivatives.

We will summarize the multiindex notation introduced in [48].

Consider $\mathbf{n} = (n_1, \dots, n_d)$ a d -dimensional multiindex vector, where the n_i entries are non-negative integers. And then define the following operators and operations:

1. Sum of components: $|\mathbf{n}| = \sum_{i=1}^d n_i = N$.
2. Factorial: $\mathbf{n}! = n_1! n_2! \dots n_d!$
3. Exponentiation of a vector $z = (z_1, \dots, z_d) \in \mathbb{C}^d$: $z^{\mathbf{n}} = z_1^{n_1} \dots z_d^{n_d}$
4. Partial derivative of a function $f(\mathbf{x}), \mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$: $\partial^{\mathbf{n}} f(\mathbf{x}) = \frac{\partial^N f(\mathbf{x})}{\partial x_1^{n_1} \dots \partial x_d^{n_d}}$

Riesz transforms can be connected to the partial derivatives [48]:

$$\mathcal{R}^{\mathbf{n}}(-\Delta)^{\frac{|\mathbf{n}|}{2}} f(\mathbf{x}) = (-1)^{|\mathbf{n}|} \sqrt{\frac{|\mathbf{n}|!}{\mathbf{n}!}} \partial^{\mathbf{n}} f(\mathbf{x})$$

where $(-\Delta)^\gamma$ is the fractional Laplace of order γ . From [48]: “Since the inverse of $(-\Delta)^{\frac{|\mathbf{n}|}{2}}$ is an isotropic low-pass-filtering operator, the net effect of the higher order Riesz transform is to extract smoothed version of the derivatives of order N of the signal of interested.”

The number of components of \mathcal{R} depends on the order of the Riesz transform N , and the dimension of the signal d :

$$M = p(N, d) = \frac{(N + d - 1)!}{(d - 1)! N!} \quad (2.11)$$

2.4.3 Generalized Steerable Framework

Simoncelli [44] coupled the multi-resolution analysis using wavelets with the steerable concept developed years before [43]. A steerable framework can be used with polar separable functions. In this case, the radial part of the function has to be calculated only once, and the polar part has only to be computed in those directions that constitute a basis of the space. After this, the

filter can be oriented, or steered to any direction, using a weighted linear combination of the basis. This approach gives a directional analysis but keeping it computationally performant.

The steerable framework can be combined with wavelets when these are isotropic, which implies polar separability. Simoncelli's framework has been used in a lot of 2D applications. Freeman in seminal work [43] wrote a generalized version for any dimension, but it was lately, Chenouard and Unser [46–48] who used rotation matrices in 3D.

Definition 2.4. A function f is steerable in three dimensions if it can be expressed as:

$$f(\mathbf{R}\mathbf{x}) = \sum_{m=1}^M k_m(\mathbf{R})g_m(\mathbf{x}) \quad (2.12)$$

where $\mathbf{x} = (x_1, x_2, x_3)$ is any 3D vector in Cartesian coordinates and \mathbf{R} is any rotation matrix in three dimensions. $\{g_m\}_{m=1\dots M}$ is the set of primary functions, ie , a basis, and $\{k_m\}_{m=1\dots M}$ is a set of interpolation functions with $M < \infty$.

2.5 Implementation Details

2.5.1 Summary

Most of the filters in this module require input images in the dual space (frequency domain). If working with regular spatial-domain images, a `itk::ForwardFFTImageFilter` has to be applied. The decision is based on performance and accuracy, avoiding expensive convolution operations and also multiple Fourier transforms.

Also, because we work in the frequency domain, we add a `itk::FrequencyShrinkImageFilter` and a `itk::FrequencyExpandImageFilter` **without** any interpolation. The shrinker chops the high frequency pixels of the image. And the expander adds zeros in the higher frequency bins. These filters require input images to be Hermitian –note that the output of a forward FFT on a real image is Hermitian.

General FrequencyExpanders and FrequencyShrinkers that can be applied to non-Hermitian complex images are not implemented, although some preliminary work can be found in <https://github.com/phcerdan/ITKIsotropicWavelets/pull/31> based on [58].

2.5.2 Frequency Iterators

Every filter that uses the frequency value of a pixel can be templated with a `itk::FrequencyImageRegionIterator`.

These kind of iterators add the functions `GetFrequencyBin()`, `GetFrequencyIndex()` to a regular `itk::ImageRegionIterator`, helping to abstract the complexity of the frequency layout into the iterator. The layout determines the order and location of the frequency bins: zero frequency –DC component–, Nyquist, low/high, or positive and negative frequencies. The layout changes depending on the parity of the image, the forward Fourier transform algorithm chosen, or if the frequencies have been shifted (`itk::FFTShiftImageFilter`).

Also these frequency iterators have data members: `FrequencyOrigin`, `FrequencySpacing`, holding metadata information about the frequency domain.

ITK puts a strong emphasis in the `ImageInformation` or metadata on spatial domain images: `Origin`, `Spacing`, `Direction`, `Index`. There are strong requirements when dealing with images with different metadata, for example, multiplication between two images is only possible when they have compatible `ImageInformation`.

Although, when applying an `itk::ForwardFFTImageFilter`, the output image, which is now in the frequency domain, still have metadata referring to the spatial domain, but nothing about the frequency origin, or spacing between frequencies.

There are three classes of `FrequencyImageRegionIterator`:

- FFTLayout: `itk::FrequencyFFTLAYOUTImageRegionIterator`

This iterator assumes that the frequency image has a “standard layout”. Different libraries, such as VNL, FFTW, `numpy.fft`, generate this layout. Each image dimension is divided in two regions, holding positive and negative frequencies.

- ZeroFrequency: The index holding the zero frequency value is at the origin $[0, \dots, 0]$ (upper-left, corner).
- Nyquist: When size N is even, the positive Nyquist frequency is located in the middle, index $N/2$, and the negative Nyquist is not stored. When N is odd, there is no Nyquist frequency, but most positive frequency is at index $(N - 1)/2$, and most negative frequency is at index $(N + 1)/2$.

The first index after the origin corresponds to the positive lowest frequency, say 0.1Hz. If this is the index 1, the latest index $N - 1$ corresponds to the least negative frequency –0.1Hz.

- ShiftedFFTLayout: [itk::FrequencyShiftedFFTLayoutImageRegionIterator](#)

The standard layout can be confusing to reason about, a common alternative is to shift the zero frequency bin to the center of the image via [itk::FFTShiftImageFilter](#).

- Regular: [itk::FrequencyImageRegionIterator](#).

This iterator is for images that were taken experimentally in the frequency domain. `GetFrequency()` here is just a wrap of `TransformIndexToPhysicalPoint`, and `GetFrequencyBin` is equal to `GetIndex`. It assumes that the image metadata refers to the frequency domain, so `FrequencyOrigin` and `FrequencySpacing` are equal to the regular `Origin` and `Spacing`. Iterators with `GetFrequency()` functions are needed for other classes in the module.

This abstraction will hopefully facilitate the implementation of further filters manipulating images in the frequency domain.

2.5.3 Wavelet Transform

See [subsection 2.3.4](#). The Wavelet Transform is templated over one of the `IsotropicWavelet` functions from [subsection 2.3.5](#) chosen by the user through a [itk::WaveletFrequencyFilterBankGenerator](#). The generator uses the function to generate an image or a filter bank.

We can compute the maximum levels that an input can accept with the `ComputeMaxNumberOfLevels`, right now the implementation needs inputs of the form s^M , where s is the scale factor chosen for performing the multiresolution, and M is an integer. Even though this condition is restrictive, we can resize any input using [itk::FFTPadImageFilter](#), or if the user pipeline involves neighbor iterators, prefer [itk::FFTPadPositiveIndexImageFilter](#), implemented in this module, that avoids setting negative indices.

The [itk::WaveletFrequencyForward](#) generates a set of coefficients generated after applying the high pass filters h_{band} for each level, and an approximation (the result of applying the cascade of low pass filter h_0).

The wavelet coefficients from the forward pyramid can be manipulated to perform further image analysis, for example edge detection, denoise, phase analysis for feature detection, etc. These extra analysis should be independent of the multiresolution framework.

After any manipulation, we can perform an inverse pyramid to reconstruct the image. Doing this requires plugging the modified wavelet coefficients into `itk::WaveletFrequencyInverse`. If the wavelet coefficients are not modified by any further analysis, we get exactly the same image as the original image that we input to the forward pyramid.

2.5.4 Riesz Transform

We have a `itk::RieszFrequencyFunction`, implementing a Generalized Riesz Function [48] of any order $N > 0$. This function receives an input $\in \mathcal{R}^d$ and outputs a vector of M Riesz Components, see Equation 2.11. To generate images from this function, we use the `itk::RieszFrequencyFilterBankGenerator`. To rotate or steer the result of the Riesz transform, use `itk::RieszRotationMatrix`. Please be aware of the use of the multiindex notation in these classes, see subsection 2.4.2.

2.5.5 Structure Tensor

Given an array of inputs, `itk::StructureTensor` [48] computes the linear combination (or direction) of inputs that maximizes the response for each location in the image. Instead of only measuring the response at the pixel of interest, it takes into account a local neighborhood. [48].

$$\mathbf{u}(\mathbf{x}_0) = \arg \max_{\|\mathbf{u}\|=1} \int_{\mathbb{R}^d} g(\mathbf{x} - \mathbf{x}_0) |\mathbf{I}_{\mathbf{u}}(\mathbf{x})|^2$$

$$|\mathbf{I}_{\mathbf{u}}(\mathbf{x})|^2 = \mathbf{u}^T \cdot \mathbf{I}(\mathbf{x}) \cdot (\mathbf{I}(\mathbf{x}))^T \cdot \mathbf{u}$$

\mathbf{I} is the required `std::vector` of input images. These images might be the output of a directional filter to an image (for example, directional derivatives from an image) or the basis of a steerable filter, such as a `RieszImageFilter`. Instead of just selecting the max response from the vector at every pixel, it uses the response over a local neighborhood, specified using an isotropic Gaussian window $g(\mathbf{x})$. This approach is more robust against noise. The user can control the radius and sigma of this Gaussian kernel. Estimation of the local orientation this way results in an eigen-system with matrix:

$$[\mathbf{J}(\mathbf{x}_0)]_{mn} = \sum_{\mathbf{x} \in \mathbb{Z}^d} g(\mathbf{x} - \mathbf{x}_0) I_m[\mathbf{x}] I_n[\mathbf{x}]$$

where I_m, I_n are input images, $m, n \in 0, N - 1$ and N is the total number of inputs. g is a Gaussian kernel.

The solution of the EigenSystem defined by \mathbf{J} are the N EigenValues and EigenVectors. The output of StructureTensor is a 2D Matrix of size $(N, N+1)$, where the submatrix (N, N) are the EigenVectors, and the last column $(N+1)$ are the EigenValues. The orientation that maximizes the response: \mathbf{u} is the EigenVector with largest EigenValue, which is the N th column of the output matrix. We can use the calculated direction \mathbf{u} to get a new image with max response from the inputs at each pixel with the function `ComputeProjectionImageWithLargestResponse()`, or any other direction from other eigen vectors with `ComputeProjectionImage`.

Also we can compare eigen values to study the local coherency of each pixel:

$$\chi(\mathbf{x}_0) = \frac{\lambda_N(\mathbf{x}_0) - A(\mathbf{x}_0)}{\lambda_N(\mathbf{x}_0) + A(\mathbf{x}_0)}$$

where $\lambda_N(\mathbf{x}_0)$ is the largest eigen value at pixel \mathbf{x}_0 , and $A(\mathbf{x}_0) = \frac{1}{N-1} \sum_{i=1}^{N-1} \lambda_i(\mathbf{x}_0)$ is the average of the other eigen values.

2.5.6 Phase Analysis

This module also implements a base class `itk::PhaseAnalysisImageFilter`, and a specialization `itk::PhaseAnalysisSoftThresholdImageFilter` that applies a soft-threshold technique to ignore low amplitude values. The example application 2.10, uses a [Monogenic signal](#) to study the local phase of each wavelet coefficient to perform multi-scale feature detection [55, 56].

2.6 A guided example:

I recommend the reader interested in extending this module looks to the tests for more usage options.

2.6.1 Input in the frequency domain.

The input for the `itk::WaveletFrequencyForward` has to be a complex image of a float/double pixel type. If the image is not in the frequency domain already, apply a `itk::ForwardFFTImageFilter` to get it.

Some FFT algorithms only work for specific image sizes, for example, the size has to be a power of two. You can apply a `itk::FFTPadImageFilter` to pad the image with zeros to reach the required size. However, this filter sets some padded areas with negative indices, which can generate problems with neighbor iterators. As a work around, this module introduces `itk::FFTPadPositiveIndexImageFilter` to avoid negative indices.

2.6.2 Choosing an isotropic wavelet

Choose a mother wavelet from the options available [2.3.5](#), and create a `itk::WaveletFrequencyFilterBankGenerator` with the type of the mother wavelet as a template parameter.

```
typedef itk::HeldIsotropicWavelet< PixelType, ImageDimension>
    WaveletFunctionType;
typedef itk::WaveletFrequencyFilterBankGenerator< ComplexImageType,
    ↪ WaveletFunctionType >
    WaveletFilterBankType;
```

2.6.3 Forward / Analysis

Perform the wavelet transform based on input levels and high-frequency sub bands. `ComputeMaxNumberOfLevels` is a static class function to calculate the max level. Currently, the size of input image has to be a multiple of two, but the implementation has the member `m_ScaleFactor` for future extension and relaxation of this constraint. Use `FFTPadPositiveIndexImageFilter` for padding the image with zeros for a valid size.

```
typedef itk::WaveletFrequencyForward< ComplexImageType, ComplexImageType,
    ↪ WaveletFilterBankType >
    ForwardWaveletType;
```

```

typename ForwardWaveletType::Pointer forwardWavelet = ForwardWaveletType::New();
forwardWavelet->SetHighPassSubBands( highSubBands );
forwardWavelet->SetLevels(levels);
forwardWavelet->SetInput(fftFilter->GetOutput());

```

2.6.4 Inverse / Reconstruction

Before applying the inverse wavelet and obtaining a reconstructed image, you want to modify the wavelet coefficients first. The design of the framework has tried to decouple both pyramids, so we can focus on algorithms that deal only the wavelet coefficients, on not with the details of the multiresolution framework.

Right now, only a phase analysis filter has been developed, but there are plenty of room for more, see [section 2.7](#). Set the same options used in the forward pyramid. And set the modified wavelet coefficients. If you don't modify them you will reconstruct the original image in the frequency domain.

Perform the wavelet forward with a suitable image:

```

#include "itkHeldIsotropicWavelet.h"
#include "itkVowIsotropicWavelet.h"
#include "itkSimoncelliIsotropicWavelet.h"
#include "itkShannonIsotropicWavelet.h"
#include "itkWaveletFrequencyFilterBankGenerator.h"
#include "itkWaveletFrequencyForward.h"
...

const unsigned int ImageDimension = 3;
typedef double PixelType;
typedef std::complex< PixelType > ComplexPixelType;
typedef itk::Image< ComplexPixelType, ImageDimension > ComplexImageType;

// Set the WaveletFunctionType and the WaveletFilterBank
typedef itk::HeldIsotropicWavelet< PixelType, ImageDimension >
    WaveletFunctionType;
typedef itk::WaveletFrequencyFilterBankGenerator< ComplexImageType,
    ↪ WaveletFunctionType >

```

```

WaveletFilterBankType;

// WaveletFrequencyForward
typedef itk::WaveletFrequencyForward< ComplexImageType, ComplexImageType,
    ↳ WaveletFilterBankType >
    ForwardWaveletType;
typename ForwardWaveletType::Pointer forwardWavelet = ForwardWaveletType::New();
forwardWavelet->SetHighPassSubBands( highSubBands );
forwardWavelet->SetLevels(levels);
forwardWavelet->SetInput(fftFilter->GetOutput());
forwardWavelet->Update();

// Result of the wavelet decomposition.
typename ForwardWaveletType::OutputsType analysisWaveletCoeffs =
    forwardWavelet->GetOutputs();

```

With the wavelet coefficients in hand (in the frequency domain), perform a decoupled filter, phase analysis, denoising, feature detection, etc:

```

// Manipulate coefficients for your purposes:
// Denoise, phase analysis, etc...
typename ForwardWaveletType::OutputsType modifiedWaveletCoeffs;
for( unsigned int i = 0; i < forwardWavelet->GetNumberOfOutputs(); ++i )
{
    ...
    aWaveletCoefficientFilter->SetInput( analysisWaveletCoeffs[i] );
    aWaveletCoefficientFilter->Update();
    modifiedWaveletCoeffs.push_back( aWaveletCoefficientFilter->GetOutput() );
}

```

And then plug back the modified coefficients to the inverse pyramid to get a reconstruction.

```

// Inverse Wavelet Transform
typedef itk::WaveletFrequencyInverse< ComplexImageType, ComplexImageType,
    ↳ WaveletFilterBankType >
    InverseWaveletType;

```

```
typename InverseWaveletType::Pointer inverseWavelet = InverseWaveletType::New();
inverseWavelet->SetHighPassSubBands( highSubBands );
inverseWavelet->SetLevels( levels );
// inverseWavelet->SetInputs( forwardWavelet->GetOutputs() );
inverseWavelet->SetInputs( modifiedWaveletCoeffs );
bool useWaveletFilterBankPyramid = true;
inverseWavelet->SetUseWaveletFilterBankPyramid( useWaveletFilterBankPyramid );
inverseWavelet->SetWaveletFilterBankPyramid(
    ↪ forwardWavelet->GetWaveletFilterBankPyramid() );
inverseWavelet->Update();

// The output of the inverse wavelet is in the frequency domain.
// Perform an InverseFFT to get a real image.
typename InverseFFTFilterType::Pointer inverseFFT = InverseFFTFilterType::New();
inverseFFT->SetInput( inverseWavelet->GetOutput() );

// Write or visualize the reconstructed output.
writer->SetInput( inverseFFT->GetOutput() );
```

2.6.5 PhaseAnalyzer

Here we show the results of the test [itkRieszWaveletPhaseAnalysisTest](#) with a couple of optical illusions images, a checker board and a Hermann grid.

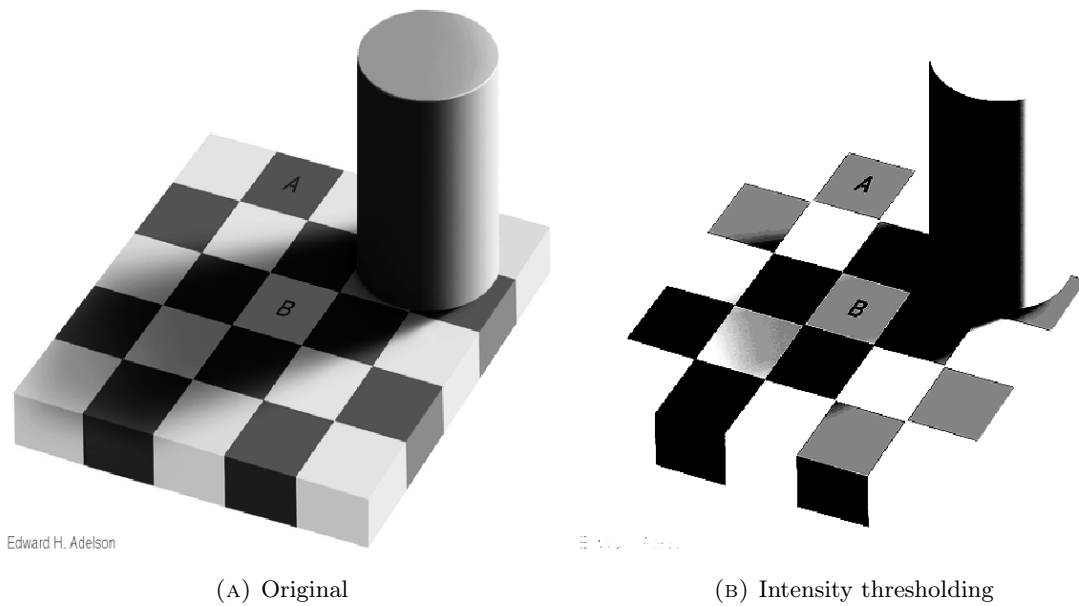


FIGURE 2.7: The original image looks like a regular check-board, but it isn't. Pixels in the regions A and B have the same intensity value (129), however our vision system performs local phase analysis that allows us to treat A,B regions as different, keeping a global checker-board structure. 2.7b uses a non-linear map of intensity-color to enhance the irregular checker board structure.

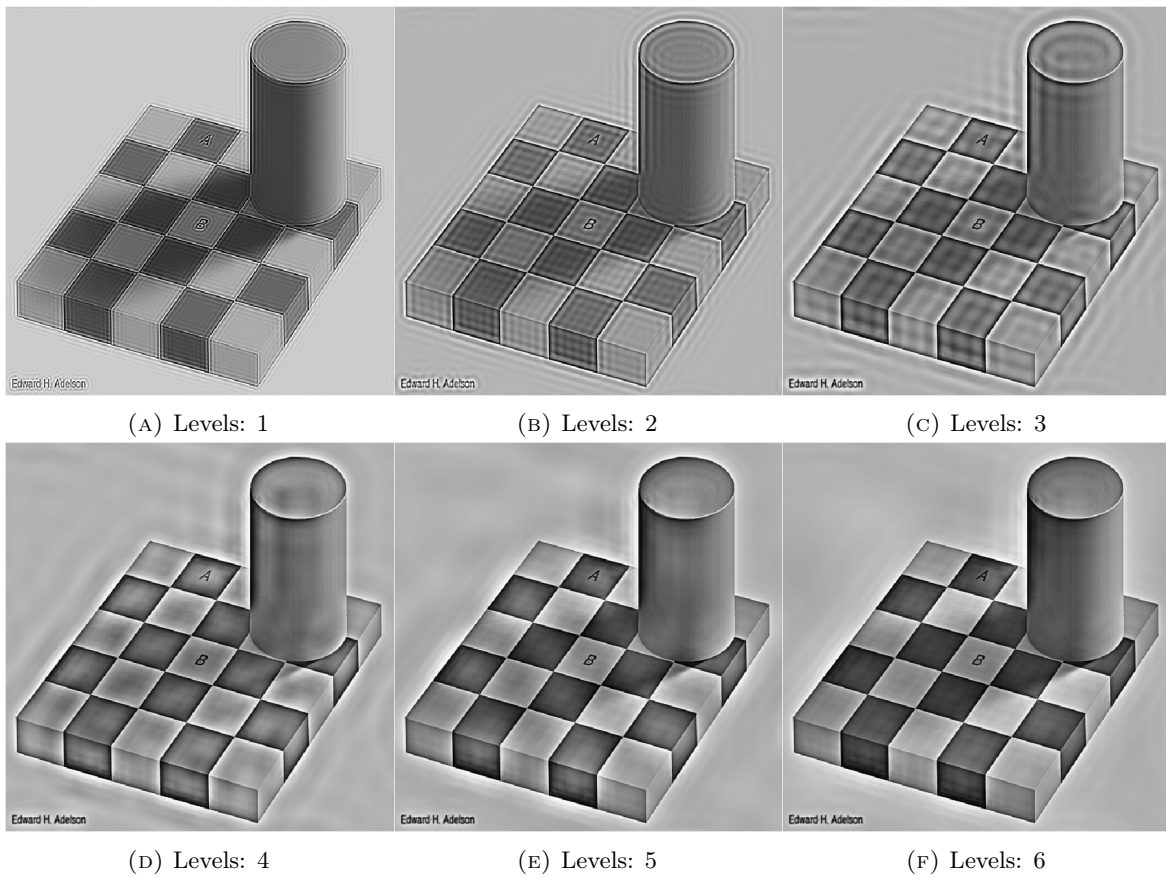


FIGURE 2.8: (Results of the phase analysis (with soft threshold) for different number of scales in the wavelet pyramid. The input image is a checker-board of size 512×512).

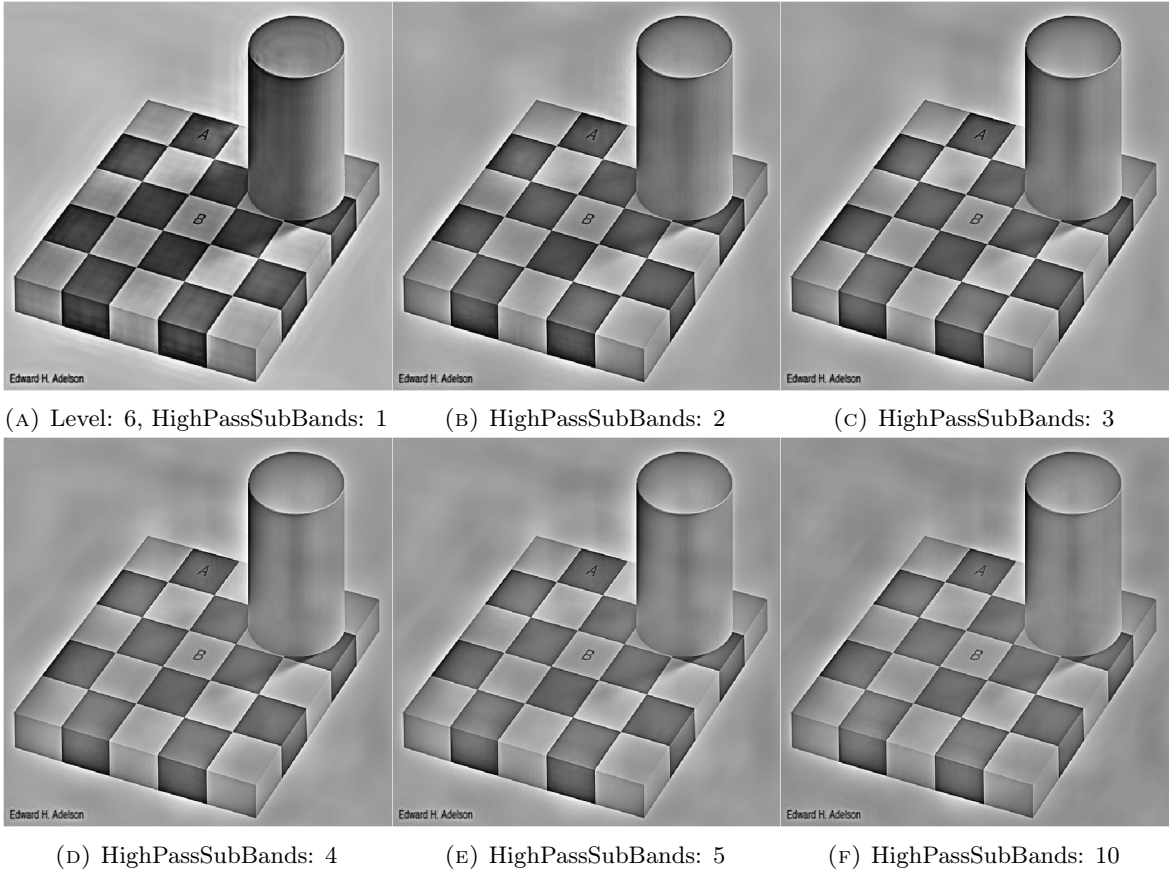
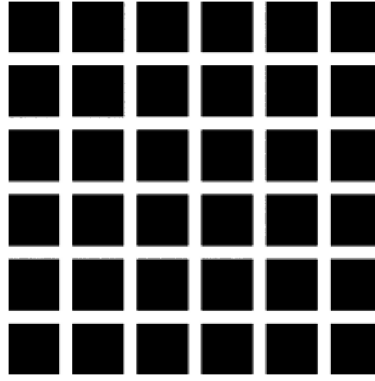
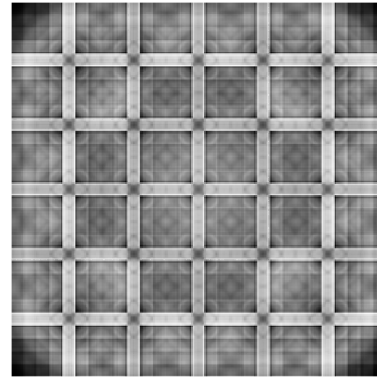
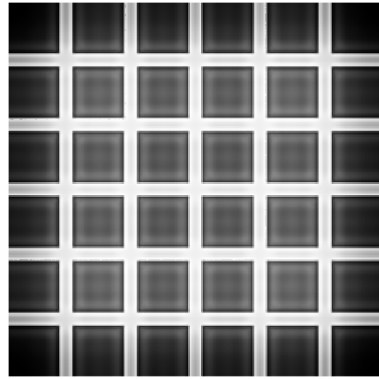


FIGURE 2.9: Using six scales (Level: 6), results for different number of high frequency sub-bands.

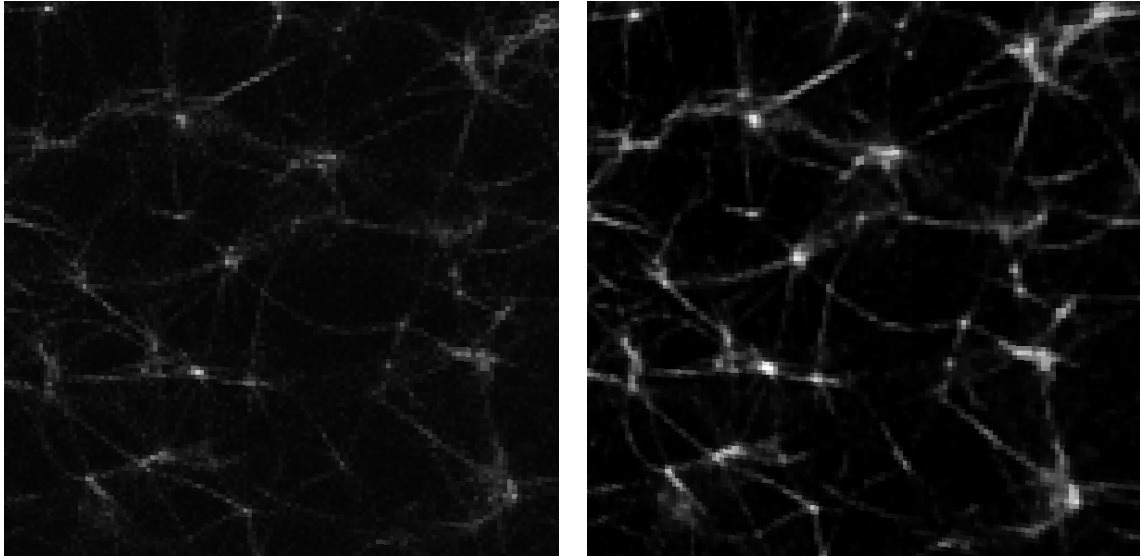


(A) Hermann Grid



(B) Levels: 8 (max), HighPassSubBands: 1 (C) Levels: 8 (max), HighPassSubBands: 10

FIGURE 2.10: The optical illusion generated by 2.10a the Hermann grid is generated by the local phase analysis of our vision. 2.10b is the result of a phase analysis with the Monogenic signal for a wavelet pyramid of 8 levels and only one high pass sub band. 2.10c is the same pyramid but with 10 high pass bands to increase the frequency resolution. Both results use Simoncelli wavelet, but there is not much difference using Held, or Vow mother wavelets.



(A) 3D image of actin biopolymer network.

(B) Wavelet phase analysis.

FIGURE 2.11: Biopolymer network (actin) 3D image enhanced and denoised using phase analysis with the Monogenic signal, with a wavelet pyramid of 4 levels and 4 high pass sub-bands using Simoncelli isotropic wavelet.

2.7 Conclusion and future work

This work provides ITK with a multiresolution analysis based on wavelet decomposition using isotropic and steerable wavelets. Also it provides utilities to work in the dual or frequency domain: Frequency iterators, shrinkers and expanders. It also provides a sub-sampler without interpolation, and a expander with zeros that work in any domain.

As an application, we showed in [Figure 2.10](#) a local phase analyzer using the wavelet coefficients based on [\[46\]](#).

Future work will be easier to implement with the tools already developed. Some work will only require plumbing operations together in a new filter, for example, creating a specific class for the Simoncelli Steerable Framework [\[44\]](#), and the more general Steerable framework using Riesz transform [\[48\]](#), that will use the already implemented [itk::RieszRotationMatrix](#).

More applications that will require extra work but use the same wavelet backbone are:

- Denoising algorithms can be developed using the exposed wavelet coefficients, for example: Gaussian Scale Mixture Models (GSM)[\[58\]](#), or SURE-LET [\[59\]](#).
- Feature detection without using template matching in an efficient way [\[60\]](#).

Implementation of these algorithms will provide state of the art solutions based on wavelets to a wider audience, and I am happy to link them from the External Repository, <https://github.com/phcerdan/ITKIsotropicWavelets> to have a reference of wavelet solutions for ITK.

Already implemented but not used in any application yet are: `itk::RieszRotationMatrix`, providing a steerable framework for the General Riesz Transform [48]. And `itk::StructureTensor`, a local estimator of the direction –steer– with the highest contribution [47].

As a note for the user trying to choose the appropriate mother wavelet, it is application dependant, but usually from the four implemented, you can assume that Simoncelli, Held and Vow will be similar when using the monogenic signal for phase analysis and bright enhancement.

2.8 Acknowledgements

I would like to thank Jon Haitz Legarreta for improving tests and documentation of the module.

3. Validation of Transmission Electron-Microscopy Imaging: Comparison with SAXS

3.1 Motivation

We were able to study a set of polysaccharide networks samples acquired by two methods, transmission electron microscopy and small-angle x-ray scattering to show that even if the preparation of the sample required for the image acquisition modified small length-scale features, network-scale information above 20nm can be trusted, as it compares well with the results obtained from unmodified samples studied with scattering.

In the spirit of open and reproducible research, the software developed for this study performing radial integration of Fourier transform of images and its visualization, is publicly available.

3.2 Abstract

Polysaccharide gels assembled from the anionic biopolymers pectin and carrageenan have been studied using transmission electron microscopy (TEM). Gels were formed in several different ways: for pectin, hydrogen bonding was used to form junction-zones between strands, while for carrageenan systems several different ion types were used to form ionotropic networks. Using this approach, several distinct network architectures were realized. In addition to preparing gelled samples for electron microscopy, a set of samples was taken without performing the additional treatment necessitated by the TEM measurements, and these were studied directly

by small-angle x-ray scattering (SAXS). Taking careful consideration of the relative merits of different image sizes and available processing techniques, the real-space images acquired by TEM were used, via radial integration of the Fourier transform, in order to produce simulated scattering patterns. These intensity-versus-wavevector plots were compared with the results of SAXS experiments carried out on the unadulterated gels using synchrotron radiation. While information regarding chain thicknesses and flexibilities was found to be modified by labeling and by changes in the dielectric constant and mechanical properties of the surroundings in the TEM, the studies carried out here show that careful protocols can produce datasets where information acquired above around 20 nm is broadly consistent with that obtained by SAXS studies carried out on unadulterated samples. The fact that at larger length scale the structure of these water-rich networks seems largely preserved in the TEM samples suggests that 3D TEM tomography experiments carried out with careful sample preparation will be valuable tools for measuring network architecture and connectivity; information that is lost in SAXS owing to the intrinsic averaging nature of the technique.

3.3 Introduction

Visualizing structures smaller than the wavelength of visible light is an important goal of benefit to a plethora of scientific disciplines, as recognized by the recent award of the Nobel Prize in Physics for developments in near-field imaging. Structural studies can be performed in real or reciprocal space. In many cases reciprocal space techniques, (scattering or diffraction), can be performed down to atomic length-scales using electrons, neutrons or x-rays. However, for some systems real-space information is paramount due to the difficulty of retrieving phase information in the reciprocal domain without extra experimental effort. Such phase information can provide valuable insights into the spatial arrangement of constituents. Spatial information is especially important when studying systems such as dispersions and polymer networks. The study of polymeric networks in soft condensed matter research typically employs real-space microscopy techniques due to the fact that the detailed spatial architecture can play a critical role in giving rise to the bulk mechanical [23, 61] and transport [62, 63] properties of the system, while scattering techniques average over heterogeneities in network connectivity.

Fibrillar proteins such as actin and collagen have been at the center of extensive studies into the underlying structure-function relationships in these complex materials [37]. Typically these systems assemble into networks on length-scales where suitably stained individual strands can

be visualized with wide-field fluorescence or confocal microscopy. Images of networks obtained in this way are generally considered to be minimally perturbed and faithful representations of the unstained samples[64]. In contrast, networks assembled from polysaccharides typically form architectures on length-scales reduced by around an order of magnitude compared with fibrillar proteins and, as such, their real-space visualization relies heavily on electron microscopy (EM) techniques. Recent technological advances have produced EM techniques that are generally considered to faithfully preserve the structures of hydrated proteins without the requirement for contrast agents [65]. However, for polymeric systems such as polysaccharides the limited phase contrast offered by these biomolecules typically requires invasive sample preparation. Even with systematic application the spectre of preparation artifacts is often raised. Despite this, few studies have been carried out in order to assess the level of confidence that can be assigned to different aspects of polysaccharide networks obtained using different EM techniques.

Previous TEM measurements have been performed on a number of different polysaccharide networks such as alginate, carrageenan and pectin[66–68] and in many of the different systems significant qualitative differences between networks can be seen, although few quantitative measurements have been performed.

In this work, several distinct network architectures were realized from two well-known gelling polysaccharides; pectin and carrageenan; key structural polysaccharides in land and marine plants respectively. In addition to preparing gelled samples for electron microscopy, a set of samples was taken without performing the additional treatment necessitated by the imaging measurements. These were studied directly by small-angle x-ray scattering (SAXS), and the results compared.

3.4 Materials and methods

Pectin: The gelation method exploited herein has been described in detail previously [69]. To summarize: a commercially available pectin sample that had a high degree of methylesterification (DM) (78%) was modified using a processive pectin methyl esterase (PME) extracted from oranges and sourced from Sigma, to obtain a block-wise sample with DM of around 40%, and the resulting polymer freeze dried. The polymeric fine structure was measured using capillary electrophoresis as described elsewhere [69]. Subsequently a

solution was made by dissolving 0.02 g of the freeze-dried pectin in 1.5 ml of deionized water. Next 0.16 g of glucono delta-lactone (GDL) was dissolved in water and mixed with the pectin solution as quickly as possible, in order to minimize any hydrolysis of GDL occurring before mixing, while being careful not to form bubbles. This solution was then loaded into the requisite sample cells for TEM or SAXS.

Carageenan: Ion-exchanged sodium kappa-carageenan samples were provided by Dupont. Relevant salt solutions (30 mM KCl or 300 mM NaCl) were first made by dissolving the required amount of salt in a volumetric flask and using millQ water. The required amount of dry carrageenan in order to produce 1% w/w solutions was then weighed out and subsequently suspended in the relevant salt solution. The solutions were then heated to 60 °C while stirring until all powder was dissolved. This solution was then loaded hot into the requisite sample cells for TEM or SAXS and gels formed on cooling to room temperature.

TEM: Gels were fixed with 0.1% Ruthenium Red and phosphate buffered 2.5% glutaraldehyde prior to dehydration in a graded ethanol series and embedding in an epoxy resin (ProCure 812, ProSciTech, Thuringowa, Australia). Fixation mechanisms for aldehydes are known to some extent from protein crystallographic studies, where in particular covalent bonding with lysine residues is employed [70], but no comparable studies exist for polysaccharides. We found that Ruthenium Red [71] was essential for maintaining sample integrity during dehydration. We observed that this was due to the formation of a skin on the gel pieces. The role of glutaraldehyde in preserving or altering structure is unknown but our observations suggest that there is no effect on the bulk structure down to the level of individual molecules, where some combination of fixation and staining changes the native morphology. Indeed, some deviation from native structure must be expected due to the use of one or more of these reagents. Sections with nominal thickness of 150 nm were cut with an ultramicrotome and post-stained with 2% uranyl acetate prior to visualisation. Previous work using tomography to acquire images from pectin gels in 3D [69] suggests that this post-staining is effective throughout the section thickness. Imaging areas were pre-irradiated for 15 minutes at low magnification prior to image acquisition. Two-dimensional micrographs were recorded on a JEOL JEM-1400 transmission electron microscope with a LaB₆ cathode operated at 120 kV under low-dose conditions with synchronised beam blanking and acquisition. Projection images were collected on a Gatan Ultrascan 2K×2K CCD with a pixel size of 0.86 nm (nominal magnification of 12 000X).

SAXS: SAXS experiments were performed at the Australian Synchrotron on the SWAXS beamline where samples were prepared in ~ 1.5 mm quartz glass capillary tubes (purchased from Hampton Research or capillarytubes.co.uk) and irradiated with 11 keV radiation. For this high-flux undulator-beamline, a slit was used to attenuate the beam and multiple (typically 10) 1 or 2 second exposures were averaged over in order to increase the signal-to-noise ratio. The sample was translated between exposures in order to prevent beam damage. Scattered radiation was collected using a Dextrus Pilatus 1M detector, with sample to detector distances of 7179 mm and 719 mm.

Image Analysis: Using the FFT of real space images in order to obtain results that are easily comparable to scattering data, (intrinsically measured in the reciprocal space), has been utilized previously with two [72] and three dimensional images [73] for different systems. For this study we utilize the Fourier transform of large two dimensional images (montages) in a quest to access larger length scales (lower scattering vectors) than would be possible using reconstructed three dimensional tomograms.

In-house software¹ written in C++11, utilizing the image analysis open source library ITK [74], was used to measure the reciprocal-space properties of the images and directly compare these to SAXS data. Following the principles of open science and reproducibility, the code and data used in this article can be found in a public repository¹ under an MPL (free) license. A terminal based executable for batch processing, and a graphical interface for visualization of the analysis process are provided, as well as instructions of how to build the code to run on different operating systems.

Fundamentally the measured pixel intensity in the images, $I(\mathbf{r})$, at position \mathbf{r} , is proportional to the difference of refractive index $I(\mathbf{r}) \propto \delta n(\mathbf{r})$ to a good approximation, so that the power spectrum $P(\mathbf{k})$ of the Fourier Transform ($\mathcal{F}\{I(\mathbf{r})\} = F(\mathbf{k})$) (Eq. 3.1) is also proportional to the scattering intensity $S(\mathbf{k})$ (eq. 3.2) obtained by light scattering experiments [72].

$$P(\mathbf{k}) = |F(\mathbf{k})|^2 = \int_v \langle I(\mathbf{r})I(\mathbf{0}) \rangle \exp(-j\mathbf{k} \cdot \mathbf{r}) d\mathbf{r} \quad (3.1)$$

$$P(\mathbf{k}) \propto S(\mathbf{k}) \propto \int_v \langle n(\mathbf{r})n(\mathbf{0}) \rangle \exp(-j\mathbf{k} \cdot \mathbf{r}) d\mathbf{r} \quad (3.2)$$

Eq. 3.1 uses the Convolution theorem $\mathcal{F}\{I\} \cdot \mathcal{F}\{I\} = \mathcal{F}\{I * I\}$, with $\langle I(\mathbf{r})I(\mathbf{0}) \rangle = \int I(\mathbf{s} - \mathbf{r})I(\mathbf{s})d\mathbf{s}$.

¹<https://github.com/phcerdan/FFTRadialIntensity>

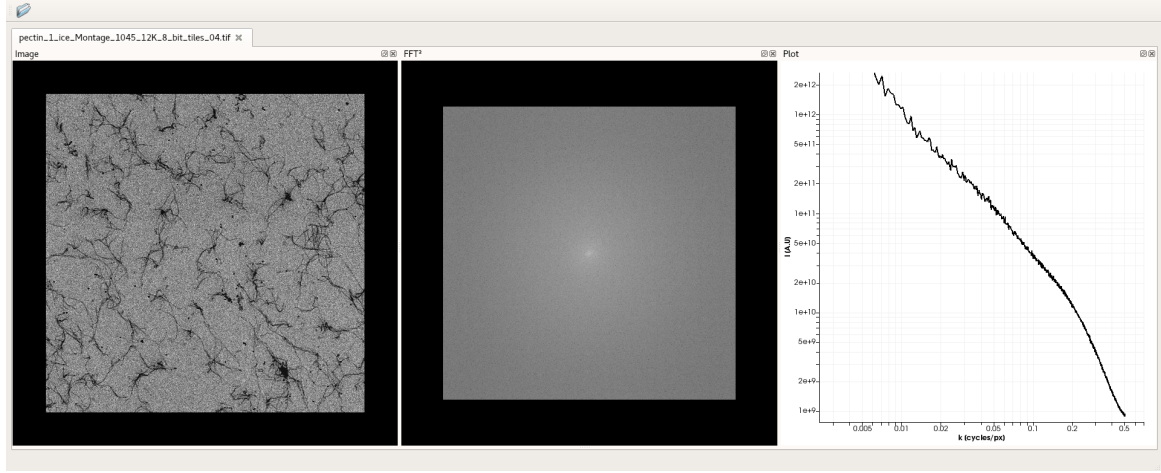


FIGURE 3.1: The Graphical User Interface (GUI) of the image analysis software developed herein. A non-interactive interface is also provided for batch processing, along with python scripts for plotting and manipulation of the generated data.

Here \mathbf{k} is the angular reciprocal space vector:

$$\mathbf{k} = 2\pi \cdot \boldsymbol{\xi} \quad (3.3)$$

Most digital Fourier Transform implementations work with non-angular $\boldsymbol{\xi}$ variables, so it is necessary to apply a factor of 2π to the image frequencies prior to comparing with the scattering vector \mathbf{q} obtained from SAXS defined by:

$$q = |\mathbf{q}| = \frac{4\pi}{\lambda} \sin(\theta/2) \quad (3.4)$$

where λ represents the wavelength of the light source and θ the scattering angle.

3.5 Results and Discussion

Figure 3.2 shows a collection of TEM images acquired from (a) an acid-induced pectin gel; and (b) and (c) carrageenan gels induced with potassium and sodium respectively, as described in the experimental section (these are montages of 5×5 separate images). In order to investigate the potential effects of the treatment required to prepare the samples for TEM, SAXS experiments were carried out on the same unadulterated gel samples with a view to experimentally measuring the structure of the biopolymer networks in reciprocal space before TEM sample preparation, albeit on a global scale, see figure 3.3. Alternatively, the Fourier transformation of the real-space images gathered by TEM after the sample preparation

(Fig. 3.2) also yields the reciprocal space structure of the samples, this time post-preparation for electron microscopy. Comparing the reciprocal space structures obtained from the real-space micrographs with those directly measured in scattering experiments allows any differences that have been introduced by the sample preparation involved in the TEM, or the exposure to ionizing radiation, to be assessed as a function of length-scale. Rather than attempting a detailed analysis of the scattering data, which typically requires empirical modeling from initial assumptions, especially when scattering curves are relatively featureless, the focus here is on the comparison of the TEM and SAXS data.

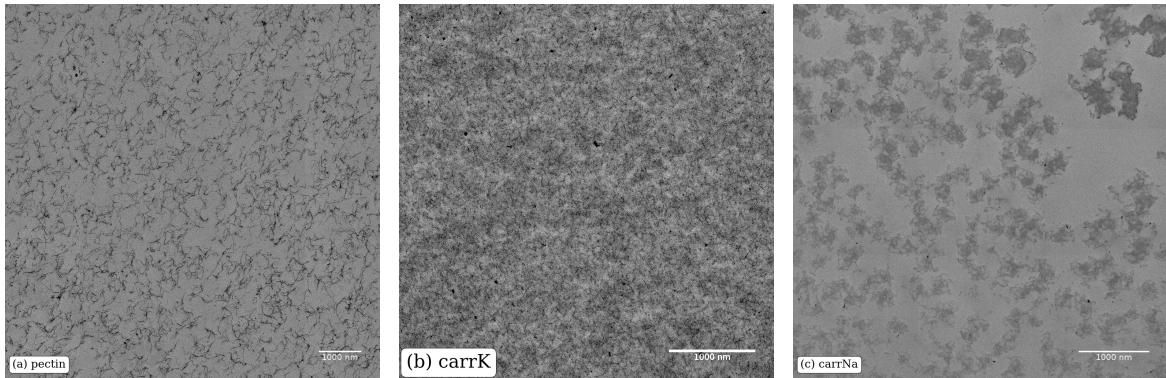


FIGURE 3.2: Montages of TEM images obtained from different polysaccharides, as described in the text.

3.5.1 Qualitative Structural Features

The most striking difference between the three TEM images is the dense cluster-like associations seen in the sodium carrageenan gel, while the other networks show architectures built from more primitive rod-like structures. The self similarity of these cluster-like associations can be quantified by applying fractal analysis to the corresponding scattering data, whereupon the slope of a log-log plot of the scattered intensity versus q reveals the average fractal dimension of the scatterers in the sample. The SAXS data is shown in figure 3.3 for the three samples together with a power-law fitting for the different regimes where self-similarity was observed. For the sodium carrageenan sample a low- q , a large length-scale, fractal dimension of (1.93 ± 0.02) , an intermediate- q fractal dimension of (2.39 ± 0.01) , and a high- q fractal dimension of (2.11 ± 0.02) were measured. The similar fractal dimensions obtained across all measurable length scales testifies to the self similarity of this network over an extended distance range in good agreement with the microscopy. In contrast, the TEM images from the pectin and the potassium carrageenan networks both exhibit rod-like features in the micrographs.

Correspondingly the SAXS data in the mid- q fractal regions do exhibit concomitant fractal dimensions of close 1, indicating the presence of rod-like features. Indeed, least-squares fitting reveals mid- q fractal dimensions of (1.27 ± 0.01) for pectin and (1.12 ± 0.01) for carrageenan respectively, with the slight deviations from unity indicating some flexibility in the rod-like regions.

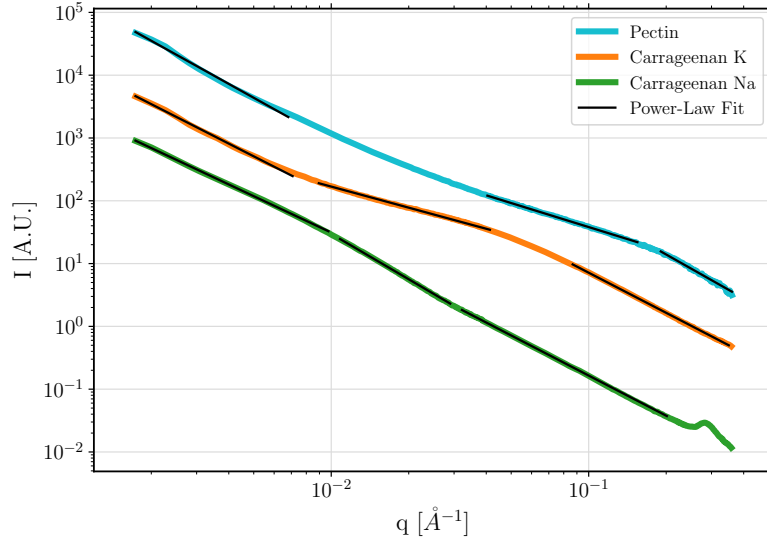


FIGURE 3.3: SAXS results for the three samples shown in the TEM micrographs of figure 3.2, in the absence of TEM sample preparation. Black lines signify power law fitting, as is described in section 3.1 of the text.

3.5.2 Consistency Between Techniques at Different Lengthscales

Figure 3.4 shows the reciprocal space data obtained by Fourier transformation of the montages presented in figure 3.2, for all three gelled systems, compared with the experimentally measured SAXS results. The absolute magnitude of the calculated data has been scaled in order to match as closely as possible that obtained from the experimental SAXS data. It should be noted that when performing the SAXS experiments, data is averaged over the scattering volume determined by the beam and cell geometry, roughly $0.2 \times 0.2 \times 1.5 \text{ mm}^3$, and that by using combinations of camera lengths and beam-line energies, scattering vectors q ranging between $[0.001, 1] \text{ nm}^{-1}$ (corresponding to approximately $[600, 0.6] \text{ nm}$) are routinely available at synchrotron facilities.

It can be seen that above ~ 20 nm there is remarkably good agreement between the results obtained for all three samples, over the range of different polymers and network architectures (even when displayed on a logarithmic scale). This suggests that the electron microscopy sample preparation method as described herein is minimally perturbing to results in this size range and that TEM provides visual and SAXS-consistent information on the architecture of polysaccharide networks. The extraction of the statistical properties of such architectures forms part of our current work and will be reported elsewhere. Below 20 nm however it appears that there are substantial differences in the SAXS and TEM-derived reciprocal space representations, at least in two of the gels. Given that the SAXS data represent the unperturbed gels, this suggests that analysis by classical transmission electron microscopy is not suitable for structural studies of polysaccharide gels at the level of filaments and their supramolecular arrangements. It can be hypothesized that such deviations are brought about by some combination of fixing, contrasting and plastic embedding of the networks, and/or changes to the resin that occur during exposure to the electron beam. The dehydration step during which the gel is exposed to increasing concentrations of ethanol leads to a deswelling of the network which, given the miscibility of ethanol and water, is believed to be isotropic in nature. It is worth pointing out that the structural distortions to the network due to beam exposure seem to be relatively small in comparison to the structural changes during dehydration.

Exposure of the epoxy-embedded section to the electron beam is known to cause anisotropic shrinkage in the x-y plane of the resin-embedded specimen section [75], while mass loss due to outgassing of the polymer resin in vacuum causes collapse of the volume orthogonal to the beam. Together, these beam-induced distortions are known to be complex, and they cannot be corrected by simple stretching factors that address only isotropic distortions. According to our measurements, the distortions observed at length scales > 20 nm appear to be remarkably isotropic, indicating that the end result of the sample preparation steps prior to imaging (dehydration, embedding, beam-induced changes) either cause negligible change to the gel structure or more probably, a change of some magnitude that is directly proportional to the native structure as shown by SAXS. This means that any quantification of the accumulated changes that occur during EM sample preparation and analysis can be applied to the final micrographs with confidence. Importantly, the same does not apply to length scales < 20 nm, where the electron-dense reagents used to provide amplitude contrast are unlikely to be a perfect approximation of the fibre structure. Indeed the specificity of Ruthenium Red

is not clear and even a single hexavalent cation has a diameter of approximately 1.13 nm [76]. Furthermore, post-staining with uranyl acetate was found to be necessary to facilitate image analysis. These observations suggest that the contrast agents are a primary source of discrepancies between TEM and SAXS at the level of the network strands.

Cryo- electron microscopy of vitrified gels would conceivably solve some of these sample preparation and imaging issues, and may yet prove useful in investigating the fine structure of polysaccharide networks; however, cryo sample preparation (which circumvents the need for chemical fixatives, dehydration and staining) must as a minimum contend with the naturally high water content of the gels, which makes vitrification challenging. Furthermore, polysaccharides are poor phase objects compared to e.g. proteins, making it difficult to generate the image signal-to-noise characteristics required for computational analysis of network fine structure. If vitrification of the gel was physically possible, cryo- negative staining would avoid all of the problems mentioned except for non-natural contrast. As reported in this study, the (predominantly amplitude) contrast generated by classical TEM is due to the stain rather than the fibers, and the validity of this technique at high resolution depends increasingly on the stain representing a faithful approximation of the fiber structure. Given the fact that vitrification of these gels is likely to be very difficult to achieve, we suggest that computational methods based on unwarping of the distorted structure (based on the movements of fiducial markers, for example), will be more useful in interrogating the fine structure with the required level of accuracy. If, however, staining artifacts contribute significantly to the inaccuracies, consideration must be given to imaging the natural phase contrast of unstained fibers.

3.5.3 Image Processing

The effect of de-noising algorithms commonly used to aid in the extraction and segmentation of network architectures, and in a range of broader image processing applications has also been investigated. In particular, the use of a total variation (TV) regularization [77, 78] and a wavelet based BLS-GSM algorithm [58, 79, 80] have been investigated. These filters are anisotropic and attempt to reduce noise present in the high frequency domain, while conserving edges of the regions of interest. The adjustable parameters that are manipulated in these algorithms, routinely referred to as λ in TV, and σ in BLS-GSM, were chosen in order to achieve the most significant reduction of high frequency speckles, while avoiding blurring or modification of the high intensity structures. Figure 3.5 shows the analysis in the

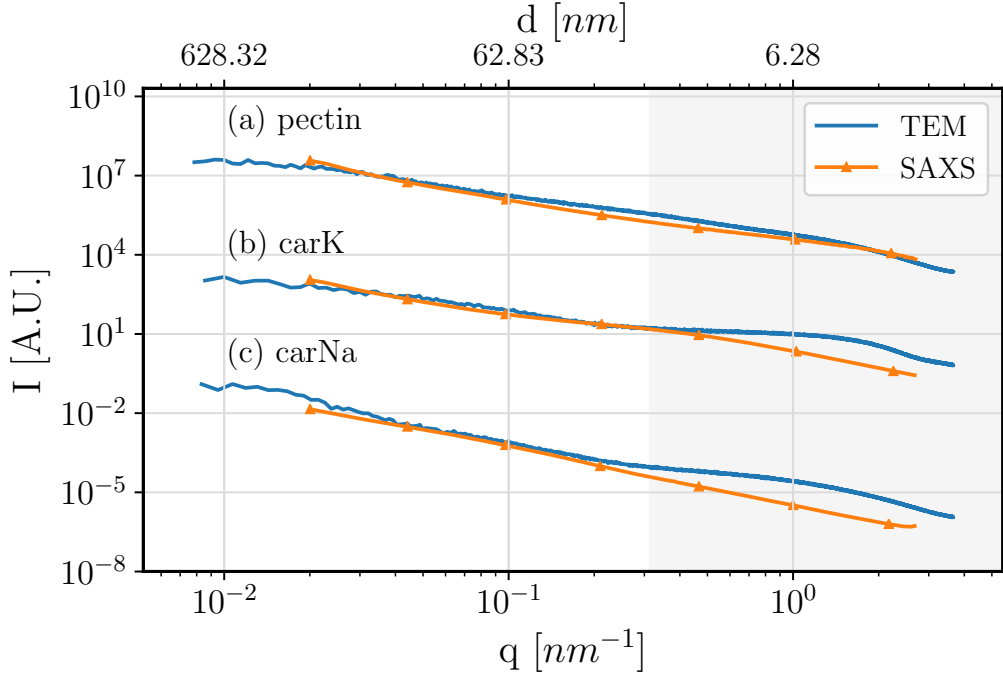


FIGURE 3.4: Comparison of data obtained from TEM best practice and SAXS. Good agreement can be seen in the large network scale (at low q s). The shaded area shows the upper q range (smaller than 20 nm or equivalently 25 pixels). The top x-axis shows the corresponding real-space length scale, where $d = 2\pi/q$.

reciprocal space of the de-noised images, and reveals that while the application of the TV algorithm in particular appears to improve the consistency of the TEM and SAXS data towards lower length-scales still, especially for the carrageenan networks, significant modification of the small-length data ($<20\text{nm}$) is produced through image processing, indicating that the information obtainable from TEM below this length-scale is not so robust and should indeed be treated with caution. On a positive note, the data in the large-length (low-frequency range) is largely unaffected by the application of these algorithms and still agrees with that recorded using SAXS. An example of the effects of de-noising in real-space is shown in Figure 3.6.

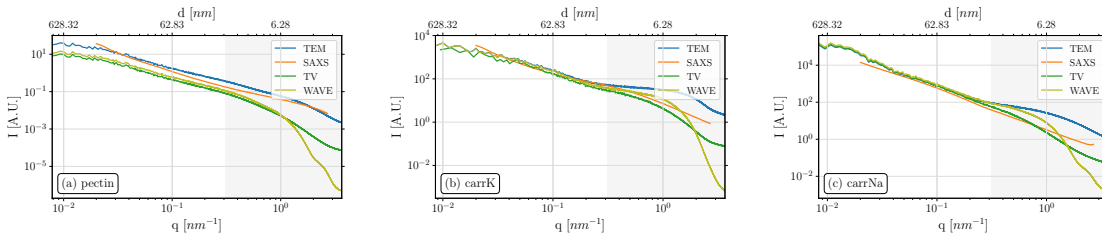


FIGURE 3.5: Results obtained when applying two widely-used de-noising techniques: total variation regularization (TV) and multi-scale wavelet denoising using a BLS-GSM algorithm.

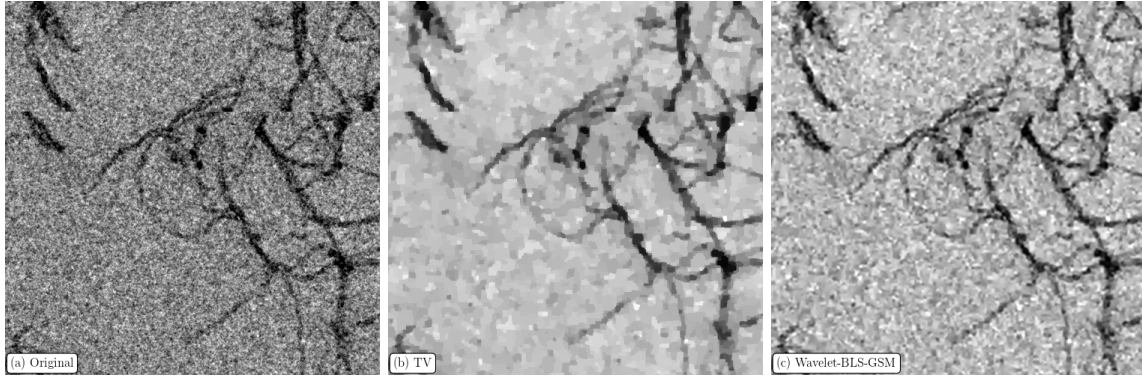


FIGURE 3.6: Visual comparison of the de-noised pectin TEM images (Fig 3.2a). (a) shows a region from the unmodified original image, (b): de-noised image using a total variation method [78] with $\lambda = 0.15$, (c) shows the same region after applying a multi-scale wavelet approach using the BLS-GSM algorithm [58] with $\sigma = 28.4$. Only high-frequency regions are affected by de-noising algorithms, as reflected in the generated I-q plots shown in Fig 3.5a.

3.5.4 Extracting Persistence Lengths

In order to investigate the possible quantitative extraction of persistence lengths from the TEM images acquired (Fig 3.2) they were analyzed directly using FiberApp [81]. This open-source software, available for tracking and analyzing polymers, filaments, biomacromolecules and fibrous objects, was used to estimate the persistence lengths of the network filaments. Figure 3.7 shows the application of FiberApp to one of the acquired TEM images and the extraction of a persistence length for the polymer chains between junction zones. While the algorithm itself works well, as demonstrated by the quality of the fit of the data to the worm-like chain model (Figure 3.7 (Insert)) a persistence length of some 140 nm is recovered. In contrast, fitting the corresponding scattering data acquired by SAXS experiments recovers a value of almost an order of magnitude less than this. (Specifically, SAXS results from such pectin acid gels have been fitted previously to a model of flexible cylinders arranged according to a fractal structure factor [69] and Kuhn lengths of around 9.75 nm were obtained.) Considering that the scattering-determined value of the persistence length of the chains in this polysaccharide network is less than ~ 20 nm, the fact that the value found in the samples prepared for TEM is different is consistent with out contention that while features of the unadulterated networks above ~ 20 nm are largely unaffected by the significant preparation procedure, smaller features can be significantly perturbed.

It can be hypothesized that such changes in the measured stiffness of the filaments are brought about by fixing, labeling and finally plastic embedding of the networks. It hardly seems surprising that in a situation where the dielectric constant of both the alcohols used for

exchange and the final embedding resin are significantly less than water, and indeed where the mechanical properties of the embedding resin when cured are significant that the original thermal fluctuations of the network fibers are now significantly hindered.

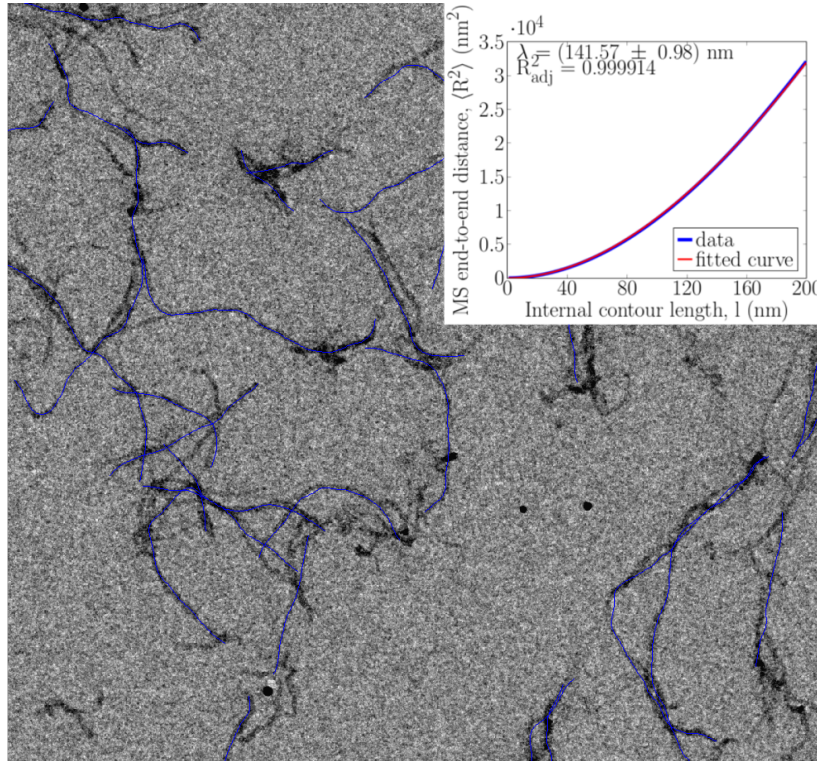


FIGURE 3.7: A TEM micrograph with FibreApp tracking output shown in blue. The insert shows the internal contour length vs mean-square end-to-end distance calculated from the corresponding tracking parameters.

3.6 Conclusions

The structures of three different polysaccharide networks have been studied over length scales from 1 nm to many microns, using SAXS and TEM imaging, revealing marked differences between gels and consistency between techniques at a resolution $> 20\text{nm}$. The networks exhibit different exemplary architectures: the sodium carrageenan gel is fractal in nature (consistent with polyelectrolytic condensation) while in contrast the potassium carrageenan and pectin samples were comprised of more homogeneous rod-like elements (consistent with the more specific interactions). The studies carried out here show that despite the fact that extensive sample preparation is needed to plastic-embed samples for TEM, careful protocols can produce datasets where information acquired above around 20 nm is consistent with that obtained by SAXS studies carried out on unadulterated samples. Information regarding chain thicknesses

and flexibilities are clearly modified by labeling and by changes in the dielectric constant and mechanical properties of the surroundings, manifesting discrepancies between the results of the techniques. Further work is planned to characterise the network structure using SAXS during the sequence of TEM preparation steps, i.e. fixation and dehydration. These experiments will elucidate the structural changes induced by the TEM sample preparation and the results will guide new strategies for sample preparations with the aim to reduce the induced structural changes and enhance the image quality, leading to reduced requirements for image processing. However, the fact that at larger length scale the structure of the pre-prepared water-rich networks seem largely preserved in the TEM samples suggests that 3D TEM tomography experiments carried out with careful sample preparation will be valuable tools for measuring network architecture and connectivity; information that is lost in SAXS owing to the intrinsic averaging nature of the technique.

The authors acknowledge the New Zealand Synchrotron group for funding for the SAXS experiments, and the MacDiarmid institute for PhD studentships (PHC, BWM), and Raffaele Mezzenga, Niklas Lorén and Annika Altskär for interesting discussions.



MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

**STATEMENT OF CONTRIBUTION
TO DOCTORAL THESIS CONTAINING PUBLICATIONS**

(To appear at the end of each thesis chapter/section/appendix submitted as an article/paper or collected as an appendix at the end of the thesis)

We, the candidate and the candidate's Principal Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of Candidate: Pablo Hernandez Cerdan

Name/Title of Principal Supervisor: M.A.K. (Bill) Williams

Name of Published Research Output and full reference:

Structural Analysis of Polysaccharide Networks by Transmission Electron Microscopy: Comparison with Small-Angle X-ray Scattering

Pablo Hernandez-Cerdan, B.W Mansel, A.Leis, L.Lundin and M.A.K Williams. "Structural Analysis of Polysaccharide Networks by Transmission Electron Microscopy: Comparison with Small-Angle X-Ray Scattering." Biomacromolecules (Jan. 2018). doi: 10.1021/acs.biomac.7b01773. ISSN: 1525-7797

In which Chapter is the Published Work: 3

Please indicate either:

- The percentage of the Published Work that was contributed by the candidate: 75 and / or

- Describe the contribution that the candidate has made to the Published Work:

The crucial analysis was enabled by code developed by the candidate. The candidate performed analysis, contributed to all aspects of the work including the writing and was well-deserved first author on the paper.

Pablo Hernandez-Cerdan
Firmado digitalmente por Pablo Hernandez-Cerdan
Fecha: 2018.04.30 19:40:10 +02'00'

Candidate's Signature

30/04/2018

Date

D453D7AC-00B4-4793-9747-45E26973BACE
Digitally signed by D453D7AC-00B4-4793-9747-45E26973BACE
University
Date: 2018.05.04 09:26:56 +12'00'

Principal Supervisor's signature

04/05/2018

Date

4. Spatial Graph Extraction: Analysis of microscopy images to obtain network architecture

4.1 Motivation

The goal of this chapter this chapter is two folds: report denoising and segmentation methods applied to 3D images of biopolymer networks, and from a binary image, obtain a spatial graph representation of the material, which encodes the connectivity and geometry of the network. The spatial graph abstraction can be used to characterize the network studying statistical distributions of different properties of the graph, but also as a direct way to encode the architecture in a format usable by computational models of network behaviour under strain.

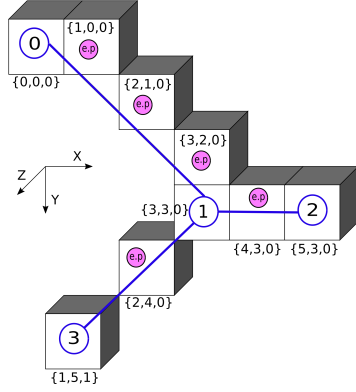
4.2 Introduction

The geometry or architecture of a biopolymer network impacts its mechanical properties and biological functions. The goal of this chapter is to explain the algorithms and software used to obtain an abstract representation, a [Spatial Graph](#), of this architecture from volumetric images taken from any microscopy technique.

A [Spatial Graph](#) (see [Figure 4.1a](#)), is a graph containing geometrical information about the image from which it is extracted. It is defined by two sets:

- A set of nodes or vertices (i.e. end-points or junctions), characterizing the spatial position of chain junctions.
- A set of edges (i.e. links), connecting pairs of those nodes. The geometric information is contained in a set of ordered edge points, reflecting the path of the connection.

The graph connectivity is described by an adjacency list [Figure 4.1b](#), with extra labels representing the position of nodes, and the set of edge points.



(A) **Spatial Graph** extracted from a set of voxels. Numbers in circles represent each node id. In brackets, the index position of each voxel.

```
graph G {
0 [index_position="0 0 0"];
1 [index_position="3 3 0"];
2 [index_position="5 3 0"];
3 [index_position="1 5 1"];
0--1 [edge_points="{1 0 0},{2 1 0},{3 2 0}"];
1--2 [edge_points="{4 3 0}"];
1--3 [edge_points="{2 4 0}"];
}
```

(B) Adjacency list representing the graph with labels containing the index position of nodes and edges points.

FIGURE 4.1: Visualization of a **Spatial Graph** (4.1a) and the corresponding adjacency list (4.1b). The graph is represented in blue, formed by nodes (voxels with id) and edges connecting them. The voxel positions (in brackets in 4.1a) of nodes and edge points are stored as labels in the adjacency list.

The following representation fully contains all the information needed to describe a network structure, as described elsewhere [30]:

1. Distribution of the distance between nodes, i.e. end-to-end distances.
2. Distribution of the fully extended length of edges, i.e. contour lengths.
3. Network connectivity, i.e. degree of nodes: representing the number of other nodes connected to it.
4. Orientation distribution. i.e. angles and direction cosines of angles between adjacent edges.

Firstly, to accomplish a reliable extraction of the spatial graph from volumetric images, the image processing analysis used to segment the biopolymer from images will be introduced, this includes the pre-processing and denoising steps in [section 4.3](#) and the binarization in [section 4.4](#).

Secondly, the thinning or skeletonization algorithm developed for creating a thin (one-pixel wide) image representation of the network in [section 4.5](#) will be presented.

Thirdly, from the resulting thin image, the abstract representation in form of the described [Spatial Graph](#) will be discussed in [section 4.6](#).

Once extracted, any [Graph](#) property can be computed from our extracted spatial graph using the tool-set of complex networks, from degree distribution to clustering coefficient. In addition also the spatial information allows us to compute geometric quantities, such as distance between nodes or angle between adjacent edges.

Finally, statistical distributions will be computed from our extracted graphs: degrees, end-to-end distances, contour lengths and direction cosines, that capture, in an analytical form, the network architecture ([section 4.7](#)).

Outline:

1. [Denoising](#)
2. [Segmentation](#)
 - [Binarization](#)
 - [Hole Filling](#)
3. [Thinning / Skeletonization](#)
 - [Critical Kernels Framework](#)
4. [Spatial Graph Extraction](#)

4.2.1 Notation

The **pixel**, **pixel position**, or **pixel index** is defined as: $\mathbf{p} \equiv (x, y, z) \in V$, where $x \in [0, X - 1]$, $y \in [0, Y - 1]$, $z \in [0, Z - 1]$ are the integer coordinates and $V \subset \mathbb{I}^3$ is the volume of the stack of images $V = \{X \times Y \times Z\}$. With Z the number of images in the stack, and X, Y the size of the $2D$ images.

Pixel value or **pixel intensity** refer to the value of the pixel in the corresponding image. The value could be an integer, a boolean, or a real number, depending on the type of image:

gray-scale, binarized or a distance map. The original **gray-scale image** I is defined as: $I(\mathbf{p}) : V \rightarrow G$ where G is a set of integers representing the gray-scale. $G = \{0, 1, \dots, G_M\} \subset \mathbb{I}$ where 0 represents black, G_M pure white, and the others correspond to different gray-scale values. $G_M = 255$ in 8 bits images, and $G_M = 2^{16} - 1 = 65535$ in higher quality images of 16 bits.

The **binarized image** B is defined as the map:

$$f_B : I \rightarrow B_T \mid$$

$$B_T(\mathbf{p}) = \begin{cases} 1 & \text{if } I(\mathbf{p}) > T \\ 0 & \text{if } I(\mathbf{p}) < T \end{cases} \quad (4.1)$$

Where $T \in G$ is the threshold value of the binarization. The threshold can also be expressed as a normalized intensity value $T_P \in [0, 1] \subset \mathbb{R}$. If $I_{max} \in G$ is the max value of I , then the threshold is $T = \text{round}(T_P \cdot I_{max})$ where the round function converts a real number into the nearest integer. The binary threshold can also be characterized with a percentage of the brightest gray-scale value involving cumulative histograms of I .

4.3 Denoising

Pre-processing the image to remove spurious noise is fundamental to obtain a good segmentation of our network in the next step.

The wavelet tools developed in [chapter 2](#) were used to perform a phase analysis [\[46\]](#) in the image. This will equalize local brightness differences, and homogenize intensity values of foreground objects. Then, an anisotropic denoising is performed, which avoids blurring information in edges of objects.

Total variation regularization [\[77, 78\]](#) methods were applied to our 3D images, showing effective denoising and maintaining sharp transitions between edges of objects and background.

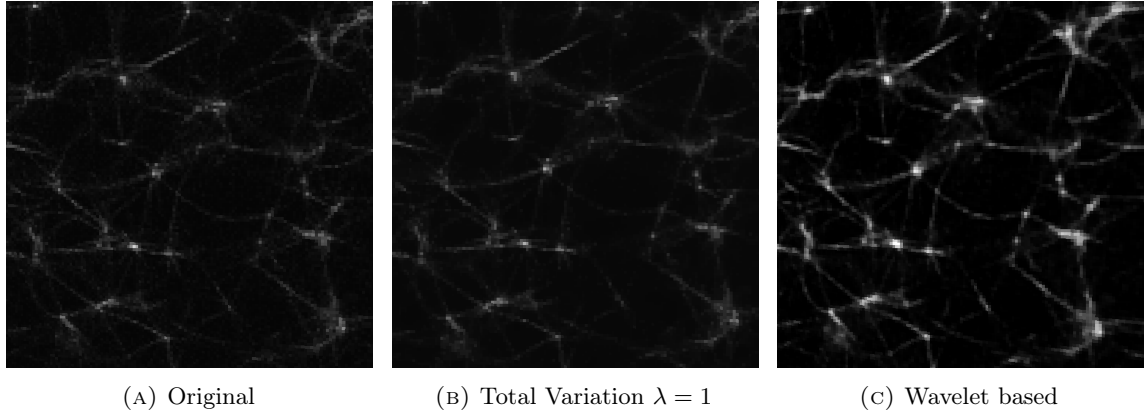


FIGURE 4.2: Pre-processing the image (actin): denoising with wavelets and total variation methods.

4.4 Segmentation

Segmentation refers to the extraction of the objects of interest from an image, discarding the rest of information. When the image is composed by many different objects, with different shapes and pixel intensities, complex approaches must be taken to reliably extract the objects of interest. For example in biology the segmentation might want to extract only cells of certain type from an image with many other actors.

In our case, our images are only composed by biopolymers, so our segmentation is easier, it reduces to a binarization task: separate background and noise from the biopolymer objects.

Easier doesn't mean easy though, grouping all the information into two clusters is inherently hard. The goal is to find an intensity value that acts as a threshold for the two groups. If the intensity along the image is not bright-normalized, it might happen that objects with absolute lower intensity are lost. This is the reason why a local approach using a region growing algorithm is preferred, where first seeds in the middle of our object of interest are selected, and then, the algorithm will grow those seeds to fill the whole region of interest based on intensity differences between neighbor pixels and the seeds.

Any binarization filter is seriously affected by noise present in the images. That is why the pre-processing and denoising step is fundamental to get a reliable result from this process.

4.4.1 Binarization

The gray levels of pixels belonging to an object are substantially different from the gray levels of the background. Thresholding is a way to separate objects from the background.

Sezgin and Sankur [82] is a good review about the subject. The output of the thresholding process is a binary image, whose one state will indicate the foreground objects, while the complementary state will correspond to the background.

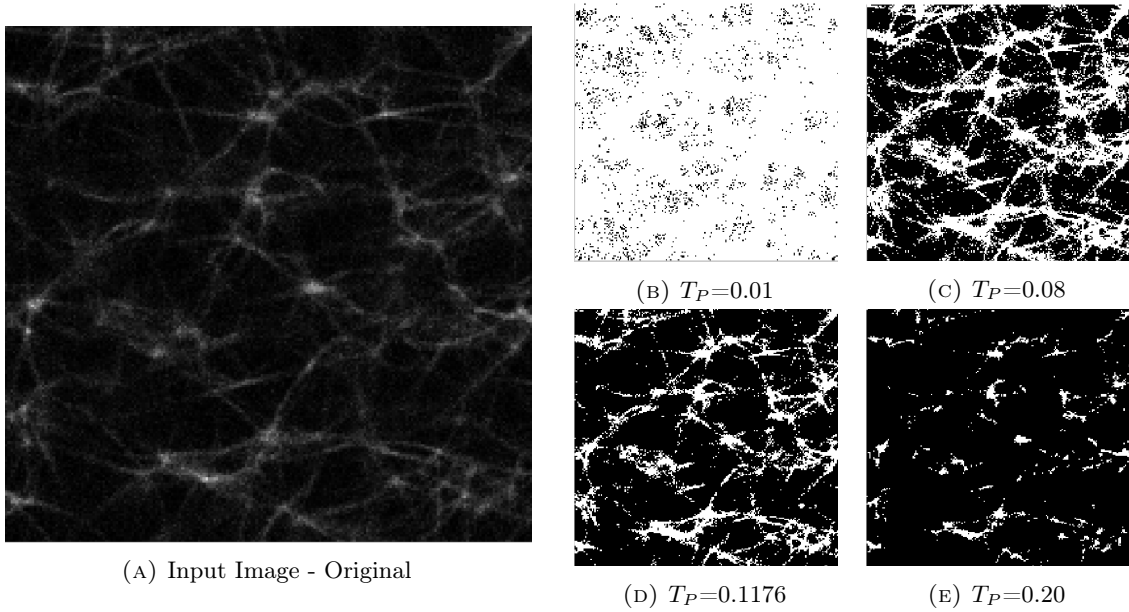


FIGURE 4.3: Different threshold values applied to a confocal image of an actin network. T_P is a normalized threshold value, if G_{max} is the brightest pixel, then $T = T_P \cdot P_{max}$. According to eq 4.1, any pixel which value is greater than T transforms to 1 (white) after the binary filter. At low values of threshold (case b), networks are blurred, but at high values (case e), some connectivity information is lost. The case d corresponds to the threshold calculated by Otsu's method.

A significant body of work already exists on thresholding. One of the most famous binary filters is Otsu's method, which minimizes the weighted sum of within-class variances of the foreground and background pixels to establish an optimum threshold [82]. This method belongs to the clustering-based methods of thresholding. Further classification of thresholding is based on:

- **Histogram shape.** Peaks, valleys and curvatures of the smoothed gray-level histogram are analyzed. *Peak-and-valley*.

- **Clustering.** The gray levels are clustered in two parts, one corresponding to the background and the other to the foreground. This method analyses separately each cluster. *Otsu, Minimum error, fuzzy.*
- **Entropy.** Exploits the entropy of the distribution of the gray levels in the image. Maximization of entropy is indicative of high information transfer. Low cross-entropy between input gray level and output binary means preservation of information. Pal, King, and Hashim [83] were the first to use Shannon entropy in image analysis. [84]
- **Object attribute.** Select the threshold based on some attribute quality between the original and binarized image. *Edge matching, connectivity, shape compactness.*
- **Spatial methods.** This class not only uses the gray distribution, also studies the dependency of pixels in a neighborhood, for example, correlation functions or co-occurrence probabilities.
- **Local adaptive methods.** A threshold is calculated at each pixel, which depend on some local statistics, like *variance*.
- **Region growing.** The binarization starts from seed pixels placed in the interior of objects wanted to be segmented. These algorithms study the neighborhood of these seeds and under some criteria, for example intensity difference, it adds the neighbor pixel to the foreground region. The process is repeated until no new pixel is added.

For our purposes, the binarization threshold will have an impact in obtaining the network. If the threshold is too high, long fibers will break in short, unconnected pieces. If the threshold is too low, fibers that are clearly separated, will become blurred together.

In Fig.4.3 the geometry of a small area of an actin network is compared after applying different binary thresholds. The stack of images of networks used here was obtained in our group by Dr. Allan Raudsepp using confocal microscopy on actin.

4.4.1.1 Region Growing Segmentation

The risk of region growing algorithms is that a seed will be planted outside our foreground objects. However, I have found planting the seeds with a normalized threshold T_p I don't have to fine tune the lower threshold too much. The results are quite resilient against small

changes of these parameters, where a pure thresholding approach will suffer a lot of variation and be prone to noise artifacts.

Using `itk::ConnectedThresholdImageFilter` with the script: [regionGrowingSegmentation.cxx](#)

Parameters:

- **Lower threshold and upper threshold:** The region won't grow into pixels outside these limits. Combined with a conservative T_p for initial seeds the results do not change with small variation of these parameters.
- **Normalized Threshold T_p for initial seeds:** Create initial seeds based on a normalized binary threshold. Taking a conservative low value, the region will grow from these seeds to fill the object.

4.4.2 Hole Filling

Due to noise, tiny holes might still be created in the binarization process. These holes (areas of OFF pixels surrounded by ON pixels) have an important effect in the following skeletonization step because they change the topology of the object and the network to be extracted. To solve it, an aggressive fill of these small holes with an iterative hole filling algorithm is realized.

Using `itk::VotingBinaryIterativeHoleFillingImageFilter` with the script: [binary denoise 3D fillholes](#)

Parameters:

- **Majority:** Number of ON pixels in the neighborhood of an OFF pixel, to switch it into ON. By default, majority = 1, this means that an off pixel will be turned on if in the neighborhood (set by radius) there are at least 50% + 1 pixels ON. If radius = {1, 1, 1} neighborhood size will be $3 \times 3 = 9$ pixels. If 5 pixels around an OFF pixel are ON, then it will be switched.
- **Neighbor Radius:** Number of neighbors to check. $R = 1$ in 3D is a $3 \times 3 \times 3$ neighborhood.
- **Iterations:** Number of iterations before stopping, the algorithm also stop if no changes are generated after last iteration.

A high value of majority and a small neighborhood is selected to only fills small holes, but let bigger empty spaces unaffected.

For 3D images, a value of:

Majority = 3, Radius = 1, Iterations = ∞ works well for our purpose of removing just small holes.

4.5 Skeletonization

The skeleton of an object is a representation of the object, which contain both, shape features and topological structures of the original object. It is widely used in computer graphics, character recognition, and analysis of biomedical images.[85–87]. The skeletonization is very sensitive to noise and boundary deformations, which generates redundant skeleton branches that disturb the topology of the skeleton graph (Fig.4.8(b)). This being the reason why the mentioned pre-processing steps of the image are important for a reliable skeletonization.

What are then the desirable properties of a skeletonization process? **a)** Preserves the topology of the original object, i.e. it has the same tails and holes. **b)** Generates a minimal or thin skeleton, i.e. the width of the skeleton is one point, only containing points which removal would alter the topology. **c)** The skeleton is centered in the object. **d)** Stable under small deformation of the boundaries (noise) of the object, not generating spurious branches.

There is no single skeletonization algorithm in the literature that fully complies with all these desirable properties, the more difficult being the robustness against rough and noisy boundaries.

The most important skeletonization algorithms can broadly be classified by:

1. Based on Voronoi diagrams [88]. These methods search the locus of centers of the maximal disks contained in the contour of the object. The skeleton can be extracted with accurate position and topology, but the time of computation is usually prohibitive, and the complicated skeleton needs to be pruned due to spurious branches.
2. Based on detecting ridges in the distance map from the boundary points. The position of the skeleton is centered, but does not guarantee to preserve the same topology than the original object [87].

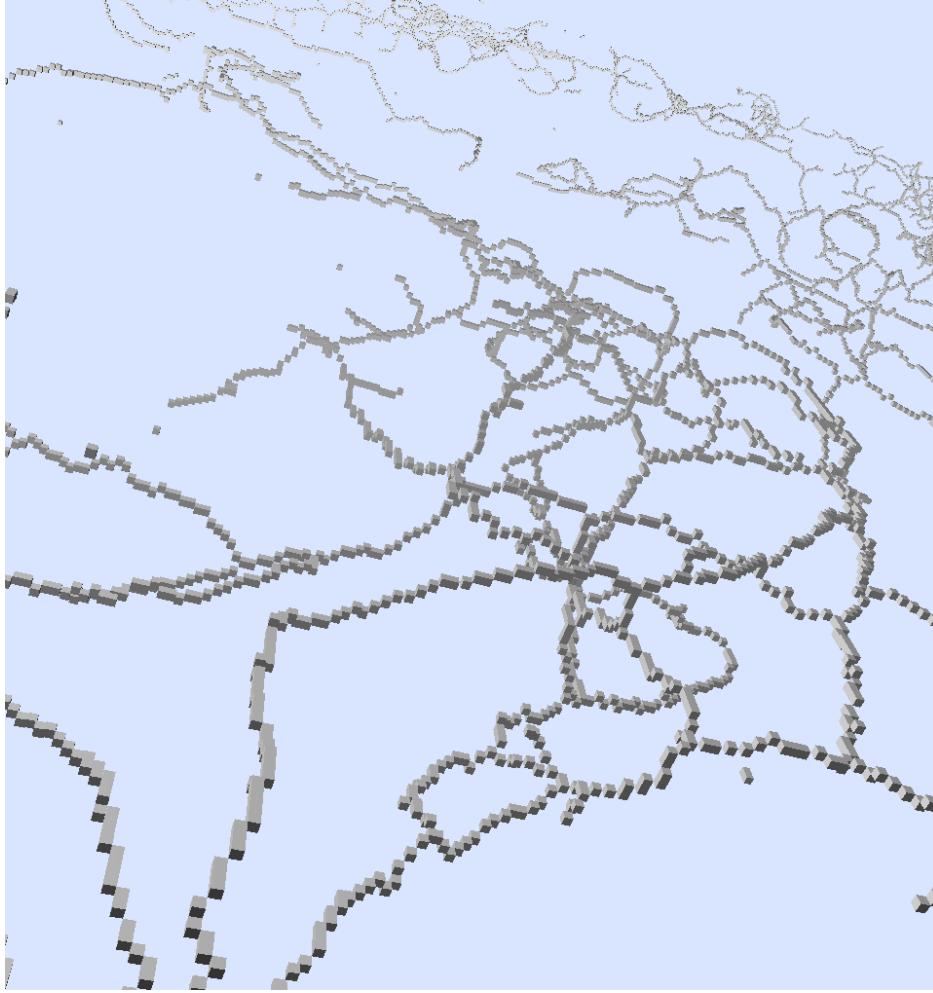


FIGURE 4.4: Thin image of a pectin network after skeletonization performed with the critical kernels framework (see [subsection 4.9.4](#)) and visualized with a 3D Viewer. The skeleton keeps the topology of the original object, is thin, centered, and robust against rough and noisy boundaries.

3. Based on iteratively peeling off points from the boundary of the object [89, 90]. The only points of the object that are deletable are simple points (a point is simple if its deletion preserves the object topology). The generated skeleton preserves the topology of the object, however is not guaranteed to be centered. The simple points can be removed sequentially or in parallel. The order in which these points are selected affects if the skeleton is thin and/or centered. However the skeleton is not stable except for small objects, usually generating many spurious branches.
4. Based on the critical kernels framework [91]. The generated skeleton keeps the same topology than the original object, is thin, and the skeleton is much more stable compared to the removal of just simple points, generating less spurious branches. The original

implementation [91] does not guarantee centered skeletons, but it can be enforced with the help of a distance map.

4.5.1 Distance maps

The distance map calculates the distance from a white pixel to the nearest black pixel. Every white pixel in the binary image has now a real value calculated using any selected distance function (euclidean, quasi-euclidean, ...). Distance maps can be calculated from a binary image, or from any image with shapes that are labeled differently. Fig.4.5 has some distance maps of actin networks. Fig.4.11 shows the values of a distance map which have been simplified from real to integer numbers for clarity.

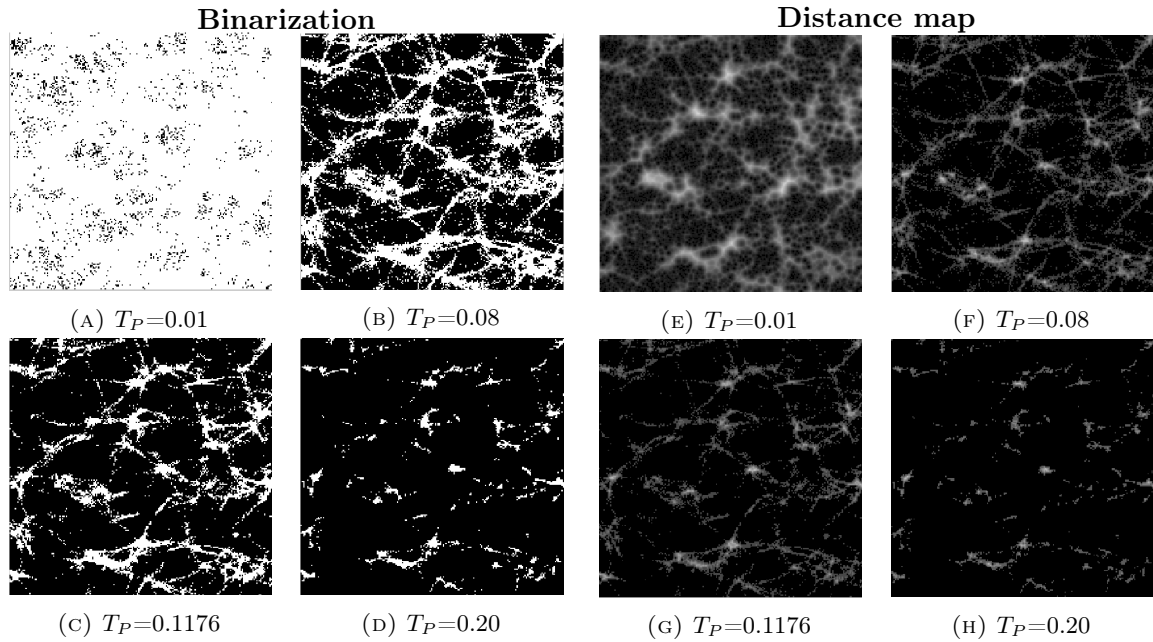


FIGURE 4.5: Example of distance maps calculated from binarized images with different normalized thresholds T_P . Images (A) to (D) show binarization images with increasing threshold and (E)-(H) show the associated distance maps. In (A) the threshold is too low, getting much of the background noise, in (D), the threshold is too high, trimming the target signal from the biopolymer.

4.5.2 Critical Kernels Framework

I contributed with this framework to DGtal <https://github.com/DGtal-team/DGtal> [92], an open-source digital topology library in c++. The work was reviewed, accepted and merged in the library since version 0.9.4 (see A.1.4 for more information).

I refer the interested reader to the original work of Bertrand and Couprie about Critical Kernels [93]. Here I will introduce a short summary of the framework, and options for the script used in this work. This summary was contributed to the [documentation of DGtal library](#).

- **Isthmus:** “Intuitively a voxel x of an object X is said to be a 1-isthmus (resp. a 2-isthmus) if the neighborhood of x corresponds -up to a thinning- to the one of a point belonging to a curve (resp. a surface).”
- **Reducible:** The complex X is reducible only if it is possible to reduce it (with thinning) to a single voxel.
- **d-cell:** A d -cell where $d \in \{0, 1, 2, 3\}$ is a pointel for $d = 0$, a linel for $d = 1$, a surfel for $d = 2$, or a spel for $d = 3$.

For example, a voxel contains 1 spel, 6 surfels, 12 linels, and 8 pointels. And the intersection of two neighbor voxels sharing a face contains 1 surfel, 4 linels and 4 pointels.

- **Clique:** Let $d \in \{0, 1, 2, 3\}$ and let $C \in V^3$ with V^3 being the collection of all voxel complexes (a finite set composed solely of voxels). C is a d -clique if the intersection of all voxels of C is a d -cell.

Any complex C made of a single voxel is a 3-clique. Furthermore, any voxel of a complex constitutes a 3-clique that is **essential** for X . Essential is the minimal clique for X .

- **K-neighborhood (of voxel S):** $K(S)$ is the set of all voxels adjacent to S . And $K^*(S) = K \setminus S$.
- **Regular Clique:** The clique C is **regular** for X if $K^*(C) \cap X$ is *reducible* to a single voxel.
- **Critical Kernel:** C is **critical** for X if C is not regular. Note that if C is a single voxel x , then C is regular for X , only if x is simple for X .

Removing simple voxels in parallel is known to change its topology, but using *critical kernels* it is ensured to keep the same topology and homotopy type after thinning.

Theorem 4.1. *The complex Y is a thinning of X if any clique that is **critical** for X , contains at least one voxel of Y .*

The details of the algorithm can be summarized in the implementation of four masks K_0 , K_1 , K_2 , K_3 , that given a d-cell (pointel, linel, surfel, spel) return a d-clique and a boolean flag with its criticality. Where K_3 corresponds to the well-known simplicity check of a voxel. An iterative thinning algorithm is then applied as shown in Figure 4.6.

Algorithm 2. PersistenceAsymThinning(X, k, p)

Data : $X \in \mathbb{V}^3, k \in \{1, 2^+\}, p \in \mathbb{N} \cup \{+\infty\}$
Result : X

```

1  $i := 0; K := \emptyset$ ; foreach  $x \in X$  do  $b(x) := 0$ ;
2 repeat
3    $i := i + 1$ ;
4    $Y := K$ ;
5   for  $d \leftarrow 3$  downto 0 do
6      $Z := \emptyset$ ;
7     foreach  $d$ -clique  $C \subseteq X \setminus Y$  that is critical for  $X$  do
8        $Z := Z \cup \{\text{Select}(C)\}$ ;
9      $Y := Y \cup Z$ ;
10   $W := \{x \in X \setminus K \mid x \text{ is a } k\text{-isthmus for } X\}$ ;
11  foreach  $x \in W$  such that  $b(x) = 0$  do  $b(x) := i$ ;
12   $W' := \{x \in Y \mid b(x) > 0 \text{ and } i + 1 - b(x) \geq p\}$ ;
13   $X := Y; K := K \cup W'$ ;
14 until stability ;

```

FIGURE 4.6: Asymmetric thinning algorithm with persistence, from Ref. [91]

Based on this framework, the concept of persistence of a voxel in the thinning [91] is implemented. The motivation is to automatically prune those branches that do not belong to a robust skeleton.

Let i be the generation (see Figure 4.6) at which a voxel x becomes an isthmus for the first time, we name it *birth date* $b(x) = i$ of the voxel. In a later generation j of the thinning process, the voxel might become deletable, we call it *death date* $d(x) = j$. The definition of *persistence* of a voxel is the difference between death and birth date, $p(x) = d(x) - b(x)$. We can choose to only keep isthmuses that have a persistence greater than a selected threshold. This persistence can be interpreted as a relative measure of the elongation of an object. With the persistence parameter, the user decides how much the elongation of an object must exceed its width to be preserved. This effectively prunes short and insignificant branches.

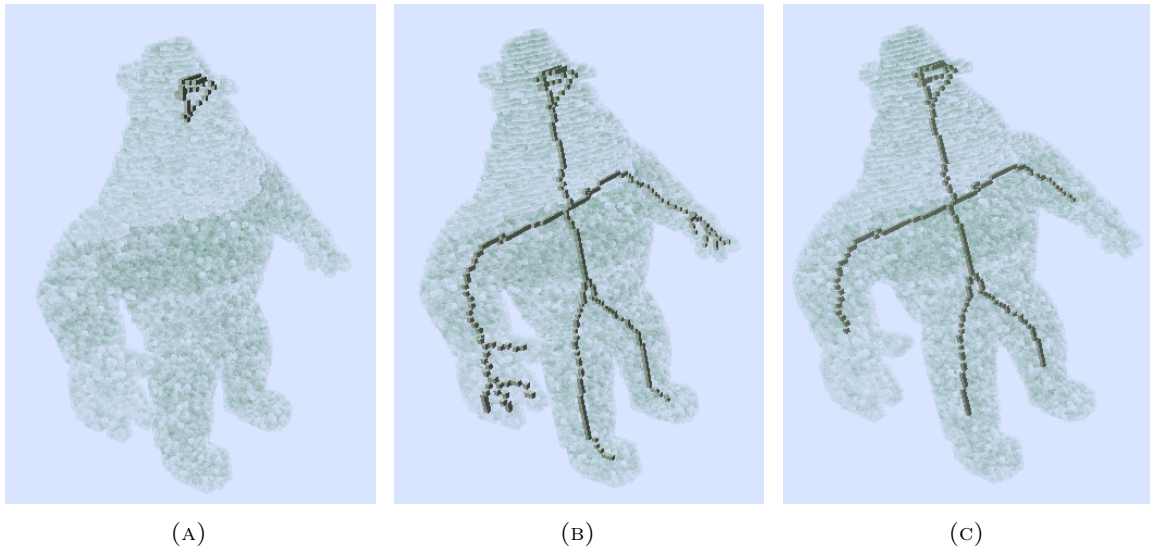


FIGURE 4.7: Asymmetric thinning of a volume with different parameters. (a): Ultimate skeleton, only voxels that conserve topology are kept. (b): keeps 1-isthmus as part of the skeleton. (c): 1-isthmus with persistence parameter $p = 10$

The implementation also allows unlimited flexibility, thanks to a functional style, to choose the *Select* and *Skel* function for any purpose.

The *Select* function chooses a voxel from a given pool. This is necessary given the parallel or asymmetric nature of the algorithm. The options are:

- **selectRandom:** Given a set of voxels (a clique), chooses one at random.
- **selectFirst:** Select the first voxel in lexicographical order.
- **selectMaxValue:** Select a voxel with the max value of another user-defined function. This wrap will be used with a distance map (see [subsection 4.5.1](#)).

The *Skel* function provides a way to keep voxels that are wanted to be preserved in the final skeleton.

- **skelUltimate:** Null, only preserve voxels whose removal alters the topology. The result of the thinning in a solid object without holes and no branches is just one voxel.
- **skelEnd:** Preserve voxels that have only one neighbor.
- **OneIsthmus:** Preserve 1-isthmus voxels.
- **TwoIsthmus:** Preserve 2-isthmus voxels.
- **skelIsthmus** Preserve 1 and 2-isthmus voxels.

- **skelWithTable** Accepts a look-up-table (LUT) of pre-computed values of certain property for certain configuration of voxels. This is used with pre-computed LUT tables for isthmus to greatly speed up the computation. It is greatly recommended and LUT tables for 26_6 (fully connected in 3D) topology are provided.

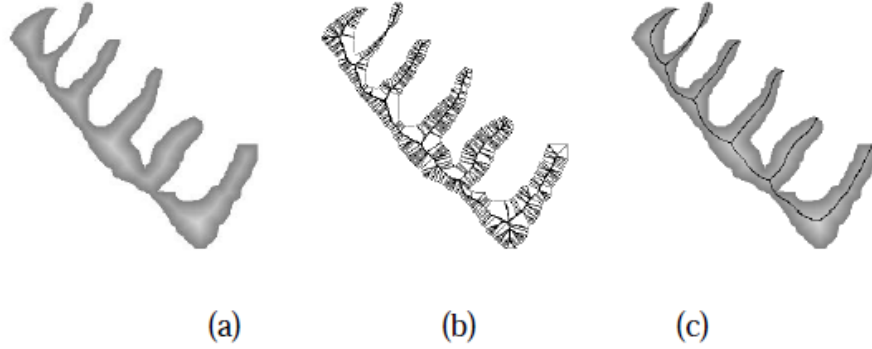


FIGURE 4.8: Skeletonization process. (a) Distance map. (b) Traditional skeleton with no pruning. (c) Skeleton generated using fixed topology methods. From Ref.[85, 86]

The original implementation [91] does not ensure obtaining a centered skeleton, but here I introduce the use of a distance map for our *Select* skeletonization function, to first select the voxels which have max value in the distance map. This selects the voxels in the medial axis, obtaining a centered skeleton while keeping the same topology. To our knowledge, this is the first time where such state of the art digital topology algorithms are applied to biopolymers images.

4.6 Spatial Graph Extraction

From the skeletonization using the Critical Kernels Framework (see subsection 4.5.2) a thin image is obtained, where all the pixels belonging to an object in the binary image are reduced to a thin representation of it, in the sense that this generated skeleton is one-pixel wide.

From here, our task is to transform this thin image or skeleton into a [Graph](#). A set of nodes connected to each other through edges. The first step, and also contributed to the DGtal library, is to obtain a **raw graph**, a graph where each node is a voxel of the skeleton, and neighbor voxels have an edge connecting them.

For regular images, this raw graph is enormous, imagine a voxel in the interior of an object, assuming a fully connected topology, this voxel would be in the center of a $3 \times 3 \times 3$ cube of voxels, having 26 connected neighbors, i.e. 26 edges or degree = 26. These neighbors voxels are nodes as well, and might be connected to others. But for thin images, where the number of voxels is greatly reduced, this allow us to start using tools from graph theory to extract a more meaningful graph, a [Spatial Graph](#).

In a [Spatial Graph](#) only nodes with degree different than 2 are considered vertices (see [Figure 4.10](#)). These end (degree = 1) and junction (degree > 2) nodes are connected to similar nodes via [Spatial Edges](#), where the bending points (degree = 2) connecting those are stored in the spatial edge as **edge points** (see [Figure 4.1b](#) for an example). The rest of vertices are kept with the same connectivity as [Spatial Nodes](#), adding its index or spatial position from the image.

This transform uses a Depth First Search ([DFS](#)) visit, starting from end-points vertices (degree=1), it follows the connected neighbors (chain-nodes of degree 2) until it reaches another end-point or another vertex with degree 3 or more. When this happens, two [Spatial Nodes](#) are added to the new [Spatial Graph](#) (the starting node source and the end node), and one [Spatial Edge](#), containing all the visited 2-degree pixels.

The [DFS](#) algorithm is applied iteratively in the graph, but the visited nodes are marked to avoid re-visiting them. After starting with end-points, it re-starts with any non-visited junction nodes (degree > 2). And to finish, self-loops are discovered, circular structures where all nodes have degree 2. Because the graph doesn't allow self-loops, edges where source and target are the same node. These self-loops are encoded into two nodes, the first is the original vertex, and the second one is created in the median of the [Spatial Edge](#). So these self loops are two nodes with two parallel edges of about the same number of edge-points between them.

The [DFS](#) algorithms is quite fast thanks to its marking of already visited nodes. There are two filters that can be applied to the process that have an impact in the generated [Spatial Graph](#).

The original **Raw Graph** can be optionally post-processed, if there are three nodes with degree 3 connected between them, the longest edges (diagonals) are removed (see [Figure 4.9b](#)).

The spatial graph gives access to the complete network geometry, and additionally allows us to analyze the architecture with sharper tools.

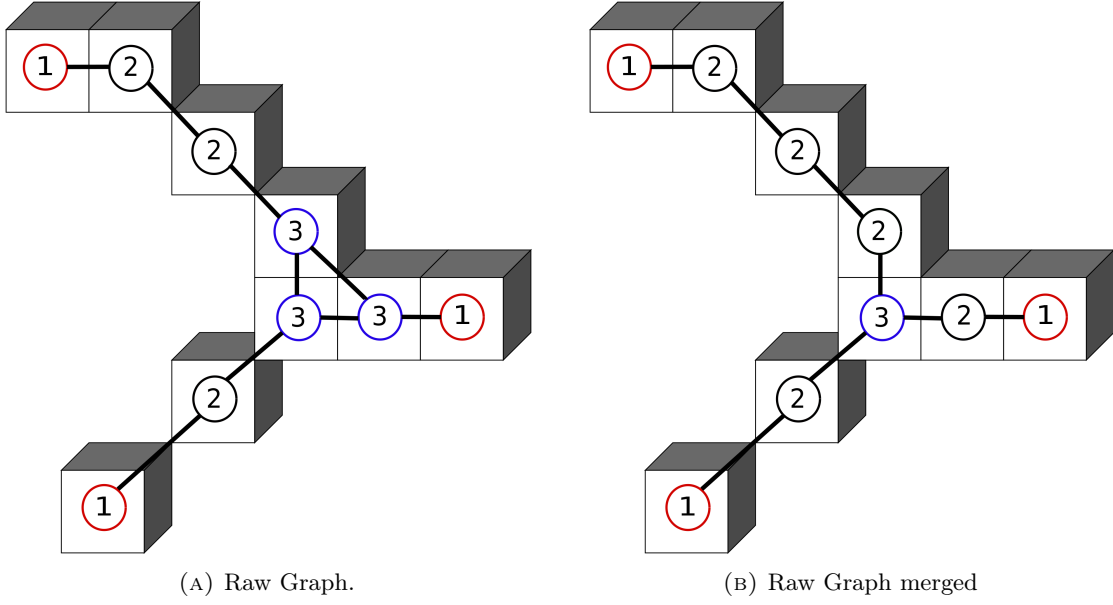


FIGURE 4.9: Extracting the raw graph from the thin image and post processed to remove the largest edges between tri-connected nodes with degree 3. Numbers show the number of edges (degree) of each node.

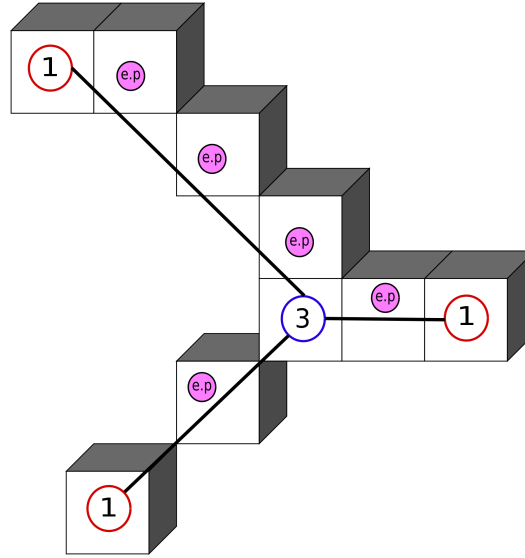


FIGURE 4.10: **Spatial Graph** where vertices with degree 2 are deleted, and its position are stored as edge points in the new edges. Numbers in circles represent degree of each node.

4.7 Statistical distributions of graph properties

With the spatial graph in hand we have at our disposal all the tool-box from the field of complex networks to characterize the network plus all the geometrical measurements that might be needed due to the storage of positions of each node and edge points in the spatial graph.

Following the approach of Lindstrom to characterize networks from statistical distributions [61, 94], the degree distribution, the end-to-end distance between nodes (ignoring edge points), the contour lengths between nodes (taking into account edge points), and the angles between adjacent edges and direction cosines of those angles are computed.

Also any existing short branch can be pruned when computing these properties, the spatial graph won't be affected, only the data extracted from it. Usually, the extremely short branches with only one or two edge points are chosen for removal to have a cleaner vision of the important branches of the network. These really short edges are not significant when getting a broad representation of network properties, but the user decides if and to what extent they are ignored.

For all our networks the degree-distribution can be described by a geometric distribution (see Equation 5.1), truncated for branch-points (only nodes with degree greater than 3) .

The end-to-end distance between nodes and the contour lengths distributions fits to a log-normal distribution, see Equation 5.2. This distribution is used to describe multiplicative processes in many fields. In biology they are used to describe growth processes [95].

These distributions are the result of the generative processes in network formation, and can be used to infer the details of such processes and patterns [96, 97].

The direction cosines of the angles between adjacent edges is fitted to a truncated power series [61], that fits well for all the networks studied, see Equation 5.3.

4.8 Testing existing thinning algorithms

Thinning algorithms have been applied to confocal microscopy images containing biopolymer protein networks before [94, 98]. In this section I test and describe the algorithms used on these extractions, and discuss why I discarded them and used the recently developed Critical Kernel Framework [99] described in subsection 4.5.2.

4.8.1 FIbeR Extraction -FIRE-

FIRE has been developed in Matlab by Andrew M. Stein [100] to extract collagen networks from confocal images. It is an open source application with a BSD licence and it can be

downloaded from <https://github.com/uw-loci/fiber-extraction>. The method [98] is based on an original idea of Wu et al. [101]. The main upgrade is the use of multiple nucleation points and the restriction that each branch extends only in a single direction from a nucleation point. In Wu et al. [101] the network has a single nucleation point at the global maximum of the image distance function D .

A Smooth filter Smooth the image with a Gaussian filter.

B Binarization threshold Keep a percentage of the brightest pixels.

C Distance map Distance map using Euclidean distance. It assigns a value to each pixel equal to the distance from that pixel to the nearest black value.

D Nucleation points NPs They are the local maxima of the distance map $D(\mathbf{u})$. The locality is defined by a box of chosen radius $s_{box} = 5$. They are only selected if they exceed a tunable threshold $\theta_{nuc} = 2$. A single nucleation point of value 4 is denoted by the thick circle in 4.11a.

E Extension of branches LMPs The next step is to trace the branches, extending from the nucleation points. To do that FIBER finds a set of Local Maximum Points (LMPs) at *each* nucleation point. LMPs occur where there is a local maximum in $D(\mathbf{u})$ on the surface of the box centered in the NP \mathbf{u} , and with radius(in pixels) the rounded up value of $D(\mathbf{u})$. See Fig. 4.11a. LMPs are then extended with other LMPs only if the direction of extension is straight enough (Fig. 4.11b). If there is no LMP with that condition, or it arrives at another nucleation point, the iteration ends.

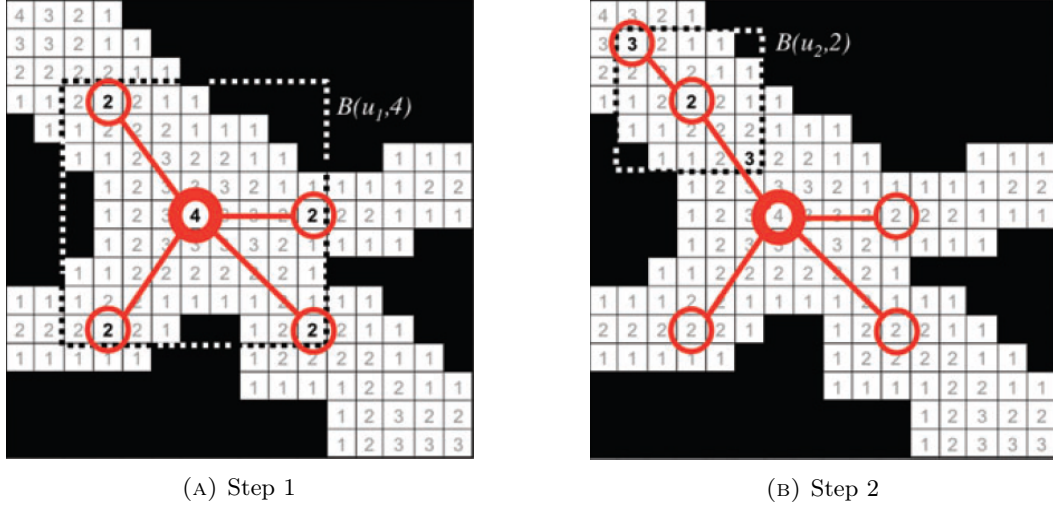


FIGURE 4.11: Tracing of the ridges of the distance map using FIRE from Ref.[98]: 4.11a Step 1: Pick a nucleation point (thick circle) and find all LMPs. 4.11b Step 2: Pick a LMP and find the next LMP which continues in the same direction. See text for details.

4.8.1.1 Results in FIRE

I have used FIRE to analyse our confocal stacks obtained from actin networks. For testing purposes, the images have been cropped due to the slowness of FIRE processing big networks, due to being in Matlab and the nature of the algorithm itself. From an original image of Actin of $V = 1024 \times 1024 \times 39$, I have cropped it to $V = 200 \times 200 \times 20$.

Fig.4.12 shows the different steps of the FIRE algorithm. In Fig.4.13 a plot is shown with the statistical distribution of end-to-end distances and degree for different thresholds T_P . These distributions will be further discussed in section 4.7 and chapter 5.

The drawback of FIRE is that the topology of the networks is not guaranteed to be conserved in the extracted skeleton. Some branches could be missed depending of its parameters, and noisy skeletons are generated. These are restrictions inherited from the algorithm originally developed in 2003 [101]. The other drawback is that is developed for Matlab which is designed for fast prototyping of algorithms, but the speed of the computations is compromised without further optimizations. Even though the software is open source, it requires that users own a license of Matlab to run the algorithm.

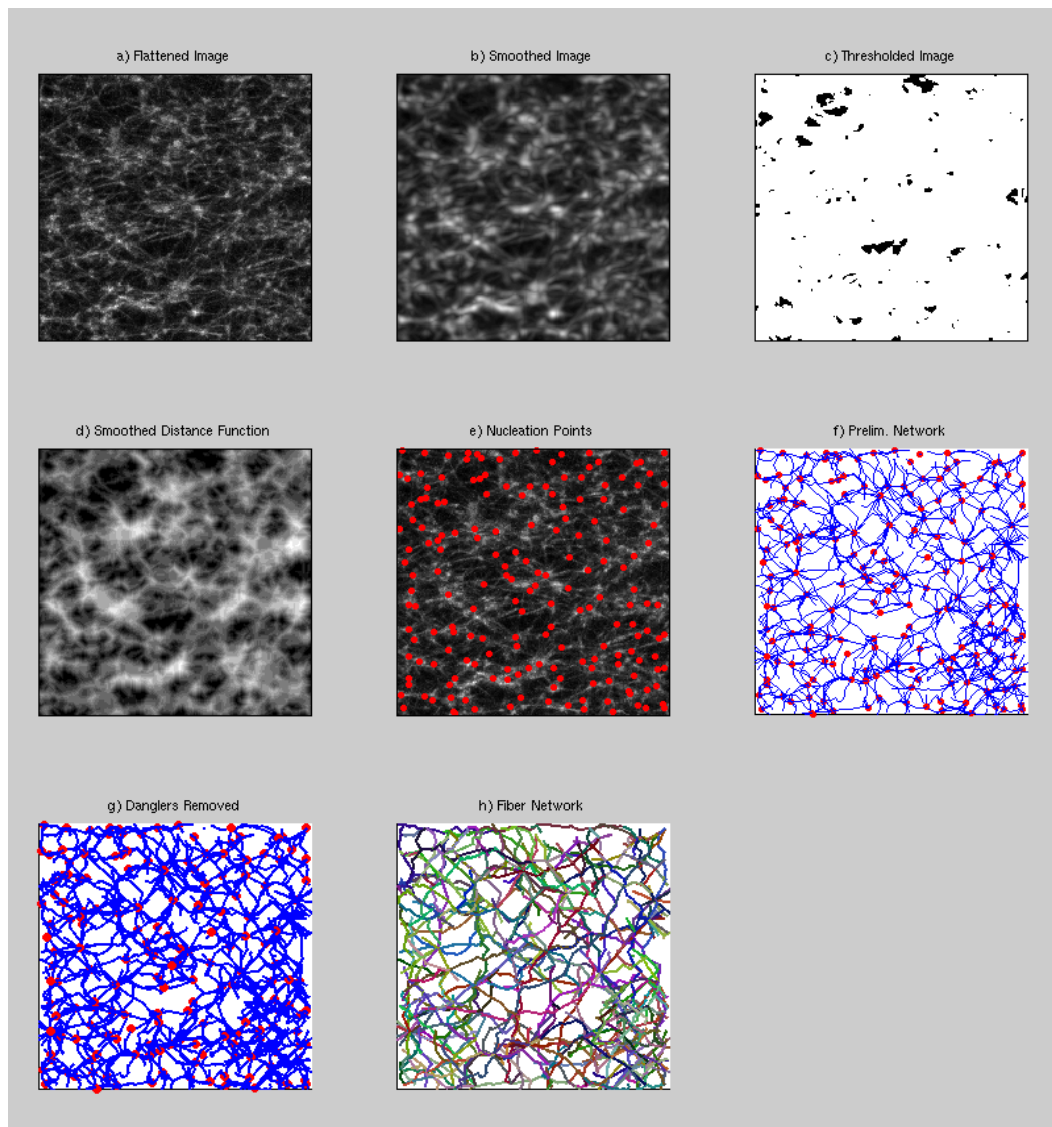


FIGURE 4.12: FIRE: step by step

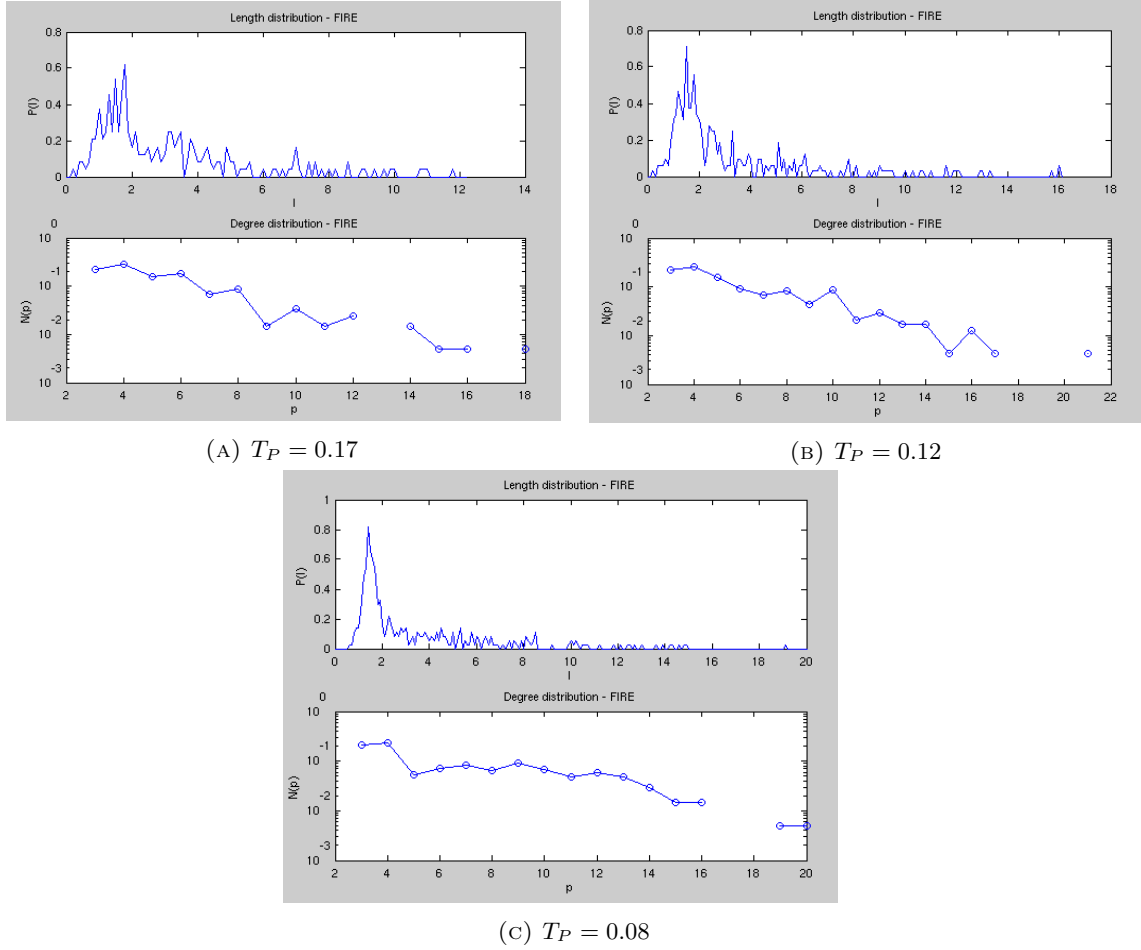


FIGURE 4.13: Extracting statistical distributions for different binary thresholds using FIRE from cropped actin networks.

4.8.2 Avizo: XSkeleton

Avizo is a software commercialized by [FEI: Visualization Sciences Group](#). Access to Avizo was obtained through a collaboration with Fonterra (Food company, NZ).

Avizo uses a Distance Ordered Homotopic Thinning (DOHT) algorithm, that ensures homotopy (i.e. keeps the same topology than the original object), thinness and centered skeletons.

XSkeleton is the package or module that used to extract the network topology. The algorithm and the result are different to FIRE. Here follows a description step by step:

A Smooth filter Smooth the image with a Gaussian filter

B Binarization threshold Multi-threshold module. This threshold is not based in percentage of the brightest pixel. It is a classical threshold turning to pure white 255 all

pixels with a value greater than the threshold value, and black all the pixels below that value.

C Distance map Distance map using Euclidean distance. It assigns a value to each pixel equal to the distance from that pixel to the nearest black value.

D Thinner module This is the **skeletonization** step. The module takes as input a label-field to be thinned and a distance map scalar field. It removes voxel by voxel from the segmented object until only a string of connected voxels remains. The thinning algorithm automatically detects dead end branches of skeleton spatial graphs. A parameter is used to distinguish them from noise on the interface of the considered regions to avoid spurious branches. Its default value is 5, i.e. the branches with a dead end which length is lower than 5 voxels are automatically considered as noise and removed. Setting this to 10, which is a rather large value, leads to only a few branches remaining in the skeleton. The drawback is that you also might miss real endpoints.

E Trace lines module Converts an image that contains lines represented by voxels into a spatial graph object.

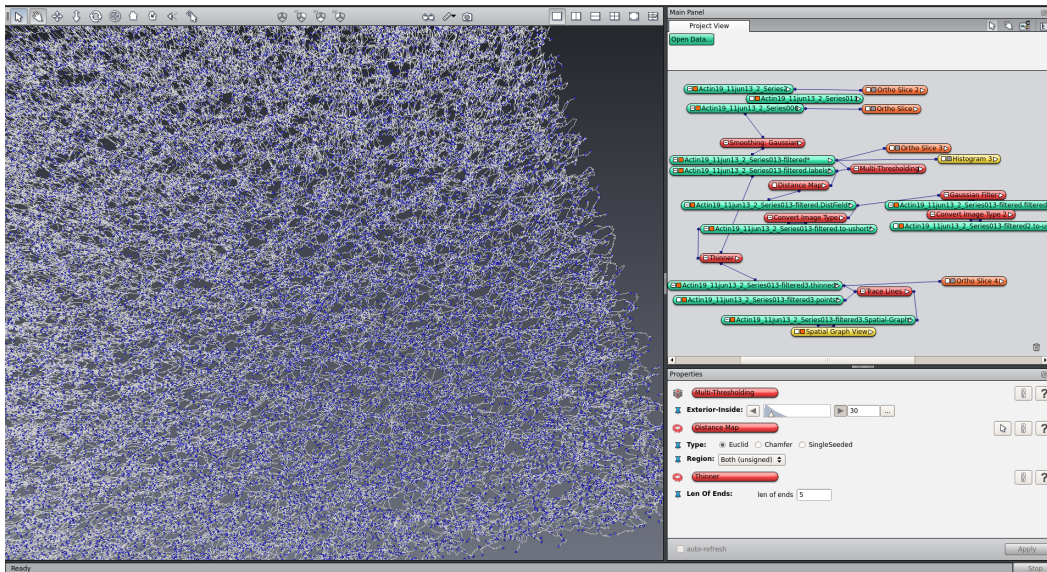


FIGURE 4.14: Avizo workspace. *Left*: Visualization of the whole network of Actin data. *Top right*: Filters applied step by step. *Bottom right*: Most influential filters parameters, binarization threshold (30 in the image) and skeletonization parameter: pruning of branches that length is lesser than the threshold (5 in the image)

4.8.2.1 Avizo results:

The whole stack of our confocal images of actin ($V = 1024 \times 1024 \times 39$) is analyzed. The key parameters in Avizo are the binarization T and the pruning threshold P .

In Fig.4.14, a regular workspace in Avizo can be seen. The different filters, the key parameters, and the result of the process, the actin network are visualized.

In Fig.4.15 the statistical distributions are extracted from the network that will be used in chapter 5 for the reconstruction of the network.

Fig.4.16 shows that statistical distributions are quite robust versus small changes in the key parameters T and P .

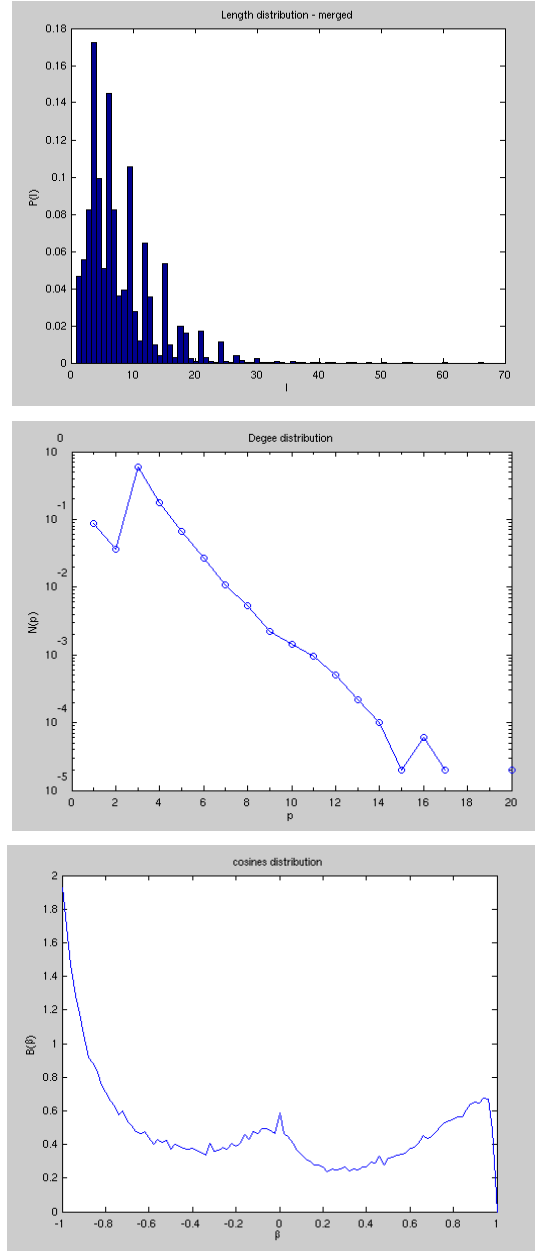


FIGURE 4.15: Actin data analyzed with Avizo with binary threshold $T = 30$ and pruning parameter $P = 5$. Length following a log-normal or exponential distribution. Degree following a geometrical distribution. Direction cosines following an unknown distribution.

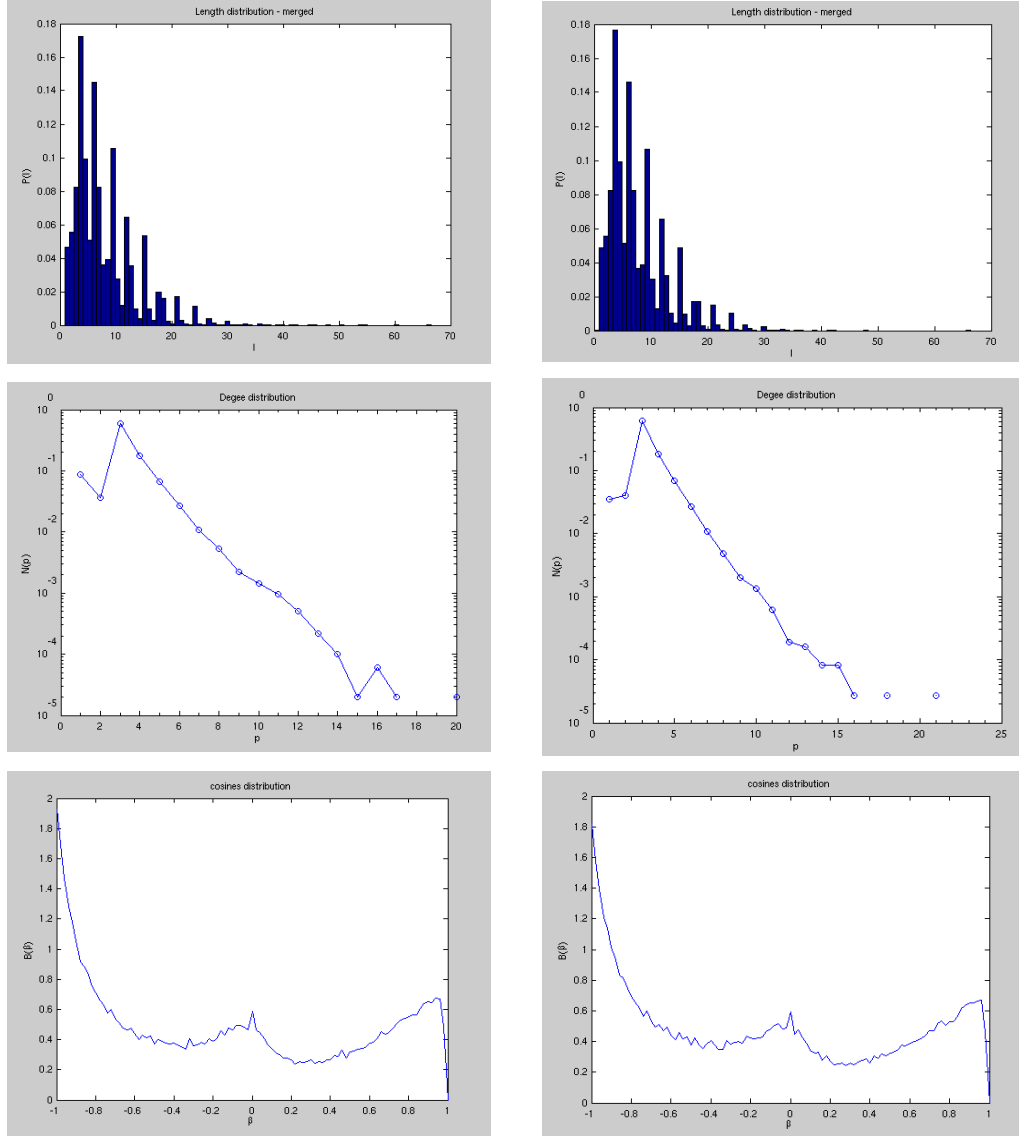


FIGURE 4.16: Actin data analyzed with Avizo with binary threshold $T = 21$ and pruning parameter: $P = 5$ in the first column, and $P = 8$ in the second column. It can be appreciated that small changes in the parameters do not have much effect much on the statistical distributions.

The parallel implementation of the skeletonization in Avizo, based on [102], makes it very fast even for big volumetric images. The main drawbacks are two: firstly the algorithm generates spurious branches in noisy contours, these can be mitigated a-posteriori with the pruning parameter P , but with the high risk of missing significant branches. Secondly, the cost of the licence for Avizo and the needed Skeletonization package is at the range of more than fifty thousands \$ per year. The exact price is not disclosed, and may vary from the packages and user cases, but it effectively makes this skeletonization only available for big institutions or industries.

4.9 Application of the Network Analysis Pipeline to Different Biopolymer Networks.

I chose to implement the critical kernel framework from [99] for biopolymer networks extraction instead of the mentioned cases to use bleeding edge algorithms from the digital topology research community in an open source format, available for any researcher to use. Being based on isthmuses instead of simple points makes it much more resilient against noise and the persistence parameter effectively removes the remaining small spurious branches. In fact, the algorithm ranks first in a quantitative evaluation of the robustness of thinning algorithms [99, 103].

The following pipeline is applied: denoising, binarization, skeletonization, extracting the spatial graph and fitting statistical distributions described in previous sections to three sets of volumetric images, obtained for three different biopolymer networks, and from the two most common 3D imaging modalities.

- **Actin:** Stack of images from Confocal Light Microscopy.
- **Carrageenan:** Image from TEM tomography.
- **Pectin:** Image from TEM tomography.

4.9.1 Confocal Light Microscopy: Actin

The actin network studied was prepared using 0.3 mg/ml actin in 40 mMol MgCl₂ and stained with phalloidin. The images were taken using Confocal Laser Scanning Microscopy (CLSM) with a Leica Instrumentat (model TCS SP5).

The volumetric image used in this study has a size of 1024×1024×106 pixels, with resolution 0.505×0.505×1.0070 μm per pixel.

The parameters used in the image pipeline:

Denoise:

Using the c++ script *rieszWaveletPhaseAnalysis* [104].

```
export WAVELET_TYPE=Simoncelli # options: Held, Vow, Shannon
export LEVELS=4
export BANDS=4
rieszWaveletPhaseAnalysis -i ${INPUT_IMAGE} -o ${OUTPUT_FOLDER} \
    -w ${WAVELET_TYPE} -l ${LEVELS} -b ${BANDS}
```

I apply an anisotropic denoise step with the itk filter `CurvatureAnisotropicDiffusion`, using the python script `Denoise_input` [105].

```
export DENOISE_ITERATIONS=20
export DENOISE_CONDUCTANCE=2.0
export DENOISE_TIMESTEP=0.0625 #default optimal value for 3D
python denoise_input.py ${OUTPUT_WAVELET} ${OUTPUT_FOLDER} \
    ${DENOISE_ITERATIONS} ${DENOISE_CONDUCTANCE}
```

Result: [Figure 4.17b](#).

Segmentation: Apply a region growing algorithm as explained in [subsection 4.4.1.1](#), using the c++ script `regionGrowingSegmentation` [106].

```
export BIN_RG_LOWER=0.0
export BIN_RG_UPPER=5000
export BIN_SAFE_PERCENTAGE=0.2
regionGrowingSegmentation -i ${OUTPUT_DENOISE} -o ${OUTPUT_FOLDER} \
    -l ${BIN_RG_LOWER} -u ${BIN_RG_UPPER} -p ${BIN_SAFE_PERCENTAGE}
```

And the hole filling algorithm described in [subsection 4.4.2](#), using the python script `Binary_denoise_3d_fillholes_iterative` [107].

```
export HOLE_MAJORITY=3
export HOLE_RADIUS=1
export HOLE_ITERATIONS=10000
python binary_denoise_3d_fillholes_iterative.py ${OUTPUT_BINARY} ${OUTPUT_FOLDER} \
    ${HOLE_MAJORITY} ${HOLE_RADIUS} ${HOLE_ITERATIONS}
```

Result: [Figure 4.17c](#).

Skeletonization

Apply a thinning using the critical kernels framework contributed to DGtal (see [subsection 4.5.2](#)) using the c++ script *Thin* [108].

```
export SKEL_SELECT=dmax
export SKEL_TYPE=1isthmus
export SKEL_PERSISTENCE=2
thin --input ${OUTPUT_SEGMENTATION} -o ${OUTPUT_FOLDER} \
  --select ${SKEL_SELECT} --skel ${SKEL_TYPE} \
  -p ${PERSISTENCE} --foreground white
```

The option *dmax* generates a distance map as shown in [Figure 4.17d](#).

Result: [Figure 4.17e](#).

Spatial Graph Extraction

The conversion from the thin image to the spatial graph representation as described in [section 4.6](#), uses the c++ script *Analyze_graph* [109].

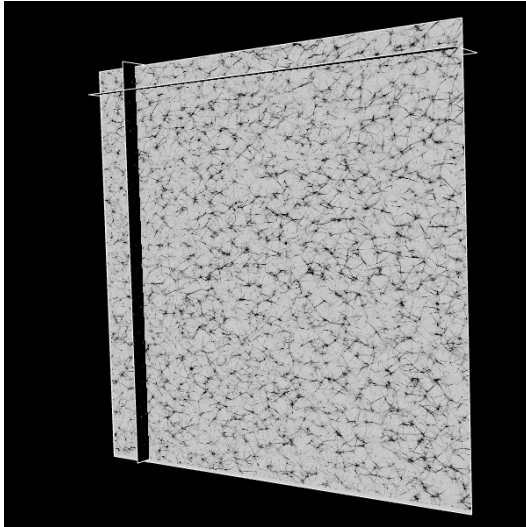
```
export GRAPH_IGNORE_SHORT_EDGES=1
export GRAPH_SPACING="5.05050745877353E-07 5.05050745877353E-07
↪ 1.00708103855232E-06"
analyze_graph --input ${OUTPUT_SKELETONIZATION} \
  --exportReducedGraph ${OUTPUT_FOLDER} --exportData ${GRAPH_DATA_FOLDER}
  --reduceGraph --removeExtraEdges --mergeThreeConnectedEdges \
  --ignoreAngleBetweenParallelEdges \
  --ignoreEdgesShorterThan ${GRAPH_IGNORE_SHORT_EDGES} \
  --spacing ${GRAPH_SPACING}
```

exportData computes the graph properties: degree, end-to-end distances, contour lengths, angle and direction cosines, and saves them into a file.

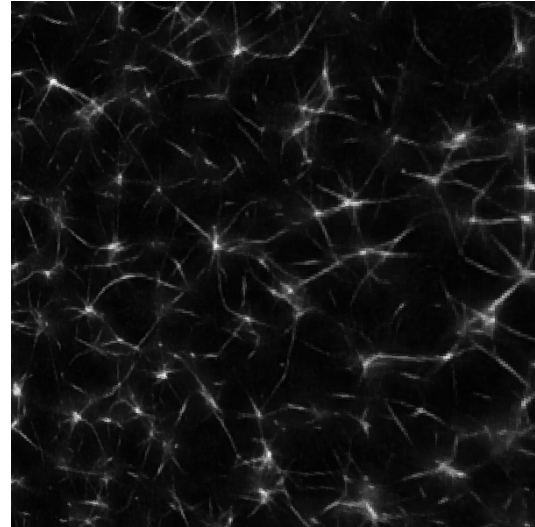
exportReducedGraph outputs the spatial graph in a .dot (graphviz) format, similar to [Figure 4.1b](#), that can be read by any graph library.

removeExtraEdges applies the removal of diagonal edges at intersections, as described in [Figure 4.9b](#).

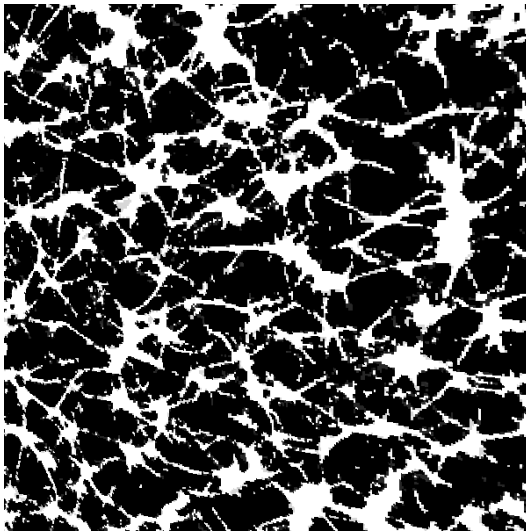
The result is a spatial graph. [Figure 4.17f](#) shows a representation of that graph without representing contour lengths, instead lines are just straight lines connecting end-to-end nodes. Node glyphs are not represented to avoid cluttering the 3D scene.



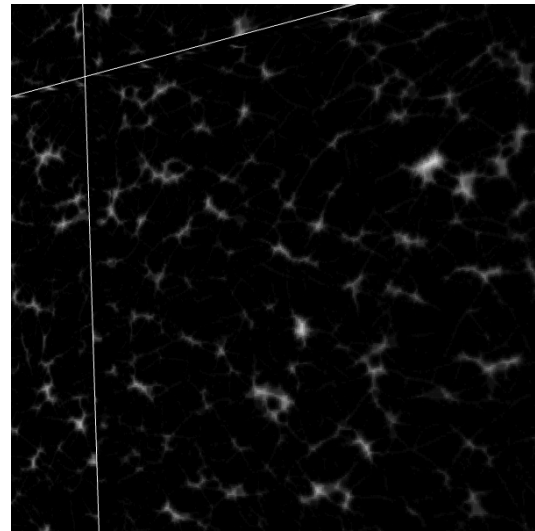
(A) Original volume.



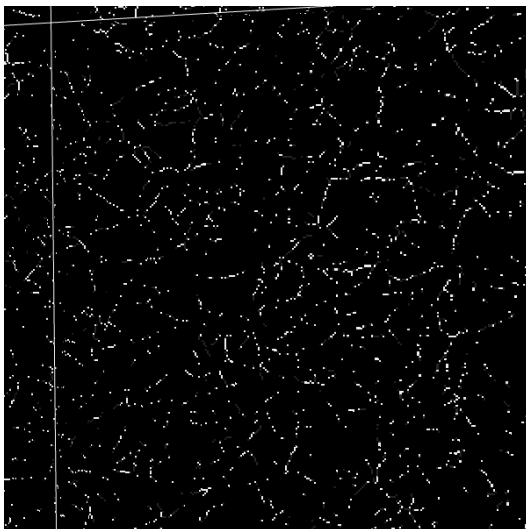
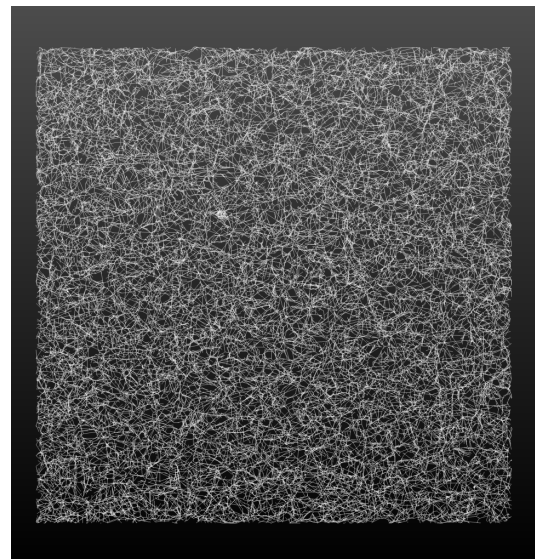
(B) Denoised.



(C) Segmentation.



(D) Distance Map.

(E) Skeletonization ($p = 2$).

(F) Graph.

FIGURE 4.17: Actin spatial graph extraction steps.

Fit graph data to distributions

I fit the graph data generated in the analysis to statistical distributions depending on the property, as described in [section 4.7](#). An estimation of the R^2 of the fit using Effron's formula [61] is shown. The degree is fitted to the mentioned truncated geometric distribution also showing the percentage of end-nodes, see [Figure 4.18a](#).

It uses the python script *Fit_to_distribution_from_data* [110] with parameters:

```
export HISTO_ETE_BINS=30
export HISTO_COSINES_BINS=21
export HISTO_CONTOUR_BINS=30
python fit_to_distributions_from_data ${GRAPH_DATA} \
    ${HISTO_ETE_BINS} ${HISTO_COSINES_BINS} ${HISTO_CONTOUR_BINS} log
```

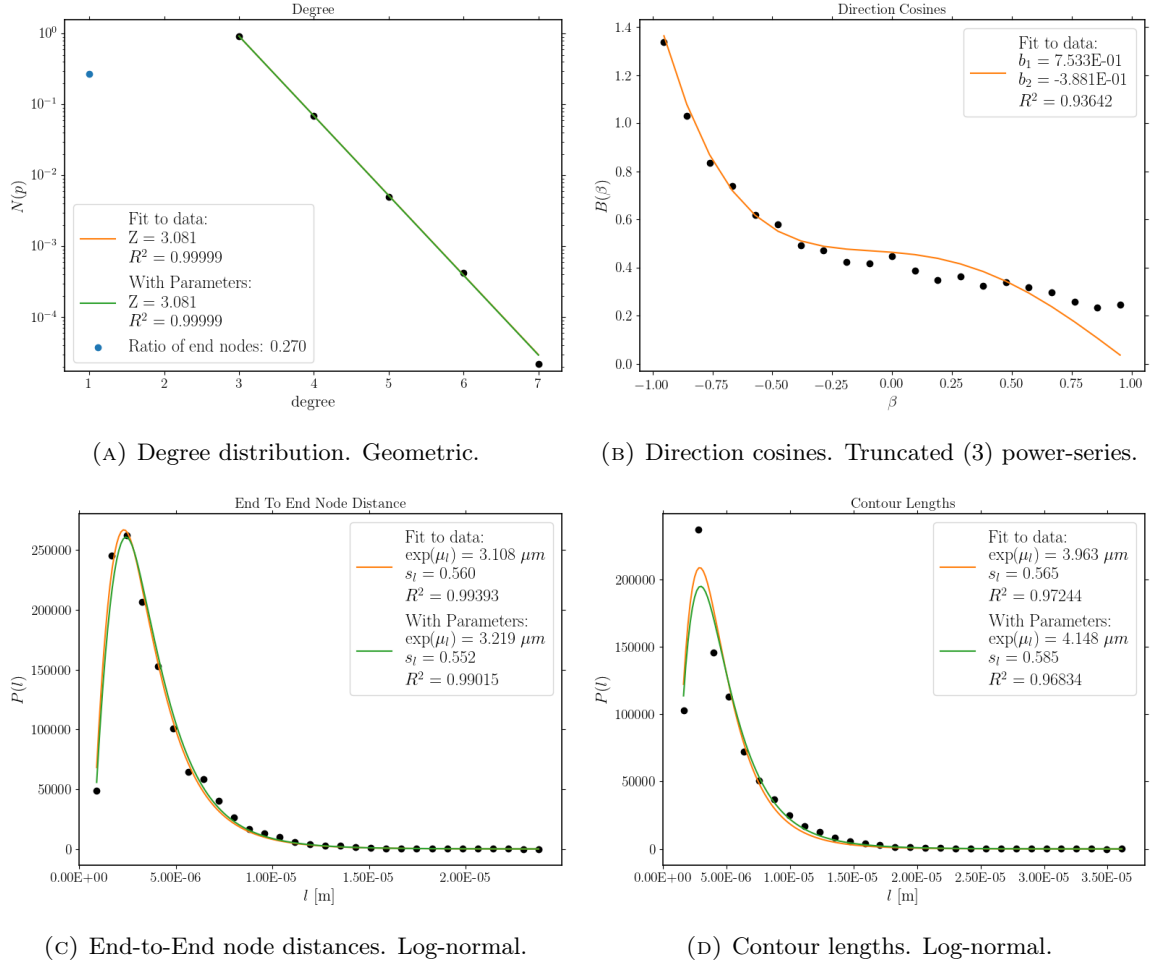


FIGURE 4.18: Statistical distributions (PDF) of actin network. Computed properties of the graph are represented in histograms with bins and normalized by area to obtain the PDF. Orange line shows the function with parameters obtained from a non-linear least squares fit to the data. Green lines show the same function with fixed parameters: 4.18a: mean degree (Z) of the network, 4.18c: mean (μ_l) and standard deviation (s_l) of logarithmic end-to-end distances, 4.18d mean (μ_l) and standard deviation (s_l) of logarithmic contour lengths. Histogram bins: end-to-end distances (30), direction cosines (21) contour lengths (30)

4.9.2 TEM preparation

The polysaccharides carrageenan and pectin are imaged with TEM tomography with the preparation published in [111] and reproduced here for completion:

Gels were fixed with 0.1% Ruthenium Red and phosphate buffered 2.5% glutaraldehyde prior to dehydration in a graded ethanol series and embedding in an epoxy resin (ProCure 812, ProSciTech, Thuringowa, Australia). Fixation mechanisms for aldehydes are known to some extent from protein crystallographic studies, where in particular covalent bonding with lysine residues is employed [70], but no comparable studies exist for polysaccharides. Ruthenium

Red [71] was found to be essential for maintaining sample integrity during dehydration. It was observed this was due to the formation of a skin on the gel pieces. The role of glutaraldehyde in preserving or altering structure is unknown but our observations suggest that there is no effect on the bulk structure down to the level of individual molecules, where some combination of fixation and staining changes the native morphology. Indeed, some deviation from native structure must be expected due to the use of one or more of these reagents. Sections with nominal thickness of 150 nm were cut with an ultramicrotome and post-stained with 2% uranyl acetate prior to visualisation. Previous work using tomography to acquire images from pecin gels in 3D [69] suggests that this post-staining is effective throughout the section thickness. Imaging areas were pre-irradiated for 15 minutes at low magnification prior to image acquisition. Two-dimensional micrographs were recorded on a JEOL JEM-1400 transmission electron microscope with a LaB₆ cathode operated at 120 kV under low-dose conditions with synchronised beam blanking and acquisition.

4.9.3 Transmission Electron Microscopy: Carrageenan

Ion-exchanged sodium kappa-carrageenan samples were provided by Dupont. Relevant salt solutions (30 mM KCl or 300 mM NaCl) were first made by dissolving the required amount of salt in a volumetric flask and using millQ water. The required amount of dry carrageenan in order to produce 1% w/w solutions was then weighed out and subsequently suspended in the relevant salt solution. The solutions were then heated to 60 °C while stirring until all powder was dissolved. This solution was then loaded hot into the requisite sample cells for TEM.

The volumetric tomography image used in this study has a size of 896×896×64 pixels, with resolution 0.86×0.86×0.86 nmper pixel.

Denoise:

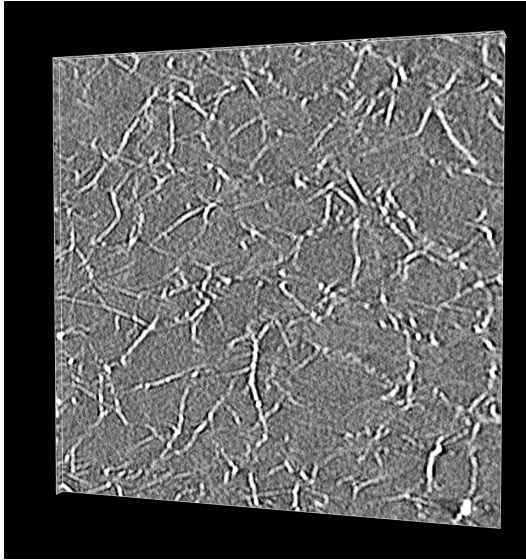
I use total variation denoising in these particular TEM images using the python script *Denoise_tv* [112], which uses proxTV library [78].

```
export TV_LAMBDA=20
python denoise_tv ${INPUT_IMAGE} ${OUTPUT_FOLDER} ${TV_LAMBDA}
```

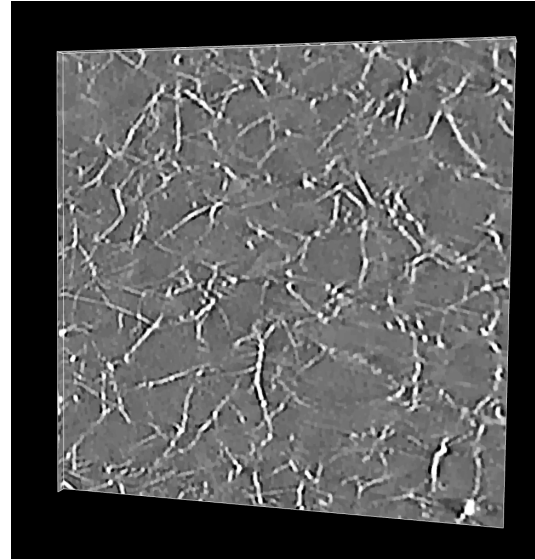
Result: [Figure 4.19b](#).

The rest of the pipeline is the same than for actin (see [subsection 4.9.1](#)), except `BIN_SAFE_PERCENTAGE`. In the Spatial Graph Extraction `GRAPH_IGNORE_SHORT_EDGES` is increased from 1 to 2, and `GRAPH_SPACING` (that represent the voxel size of the image) is changed accordingly to the metadata of the original image. The histogram bins of distances (end-to-end and contour lengths) have been increased from 30 to 50.

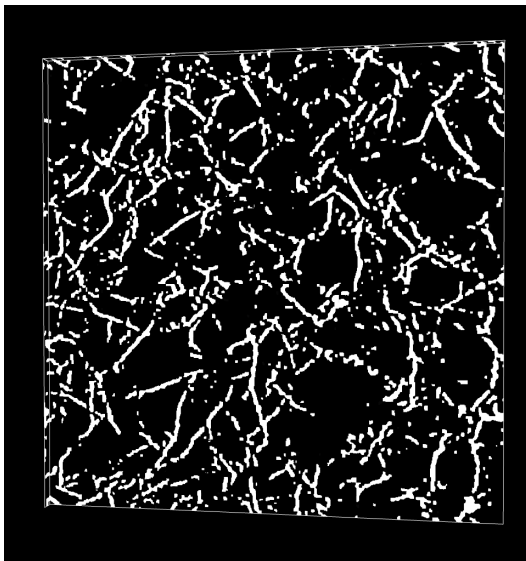
```
## Region Growing Binarization
export BIN_RG_LOWER=0.0
export BIN_RG_UPPER=5000
export BIN_SAFE_PERCENTAGE=0.05
## Hole Filling
export HOLE_MAJORITY=3
export HOLE_RADIUS=1
export HOLE_ITERATIONS=10000
## Skeletonization
export SKEL_SELECT=dmax
export SKEL_TYPE=1isthmus
export SKEL_PERSISTENCE=2
## Spatial Graph Extraction
export GRAPH_IGNORE_SHORT_EDGES=2
export GRAPH_SPACING="0.86E-09 0.86E-09 0.86E-09"
## Fit graph data to distributions
export HISTO_ETE_BINS=50
export HISTO_COSINES_BINS=21
export HISTO_CONTOUR_BINS=50
```



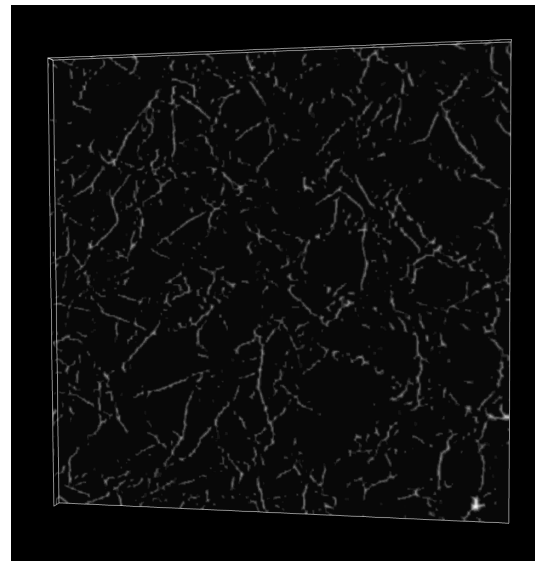
(A) Original volume.



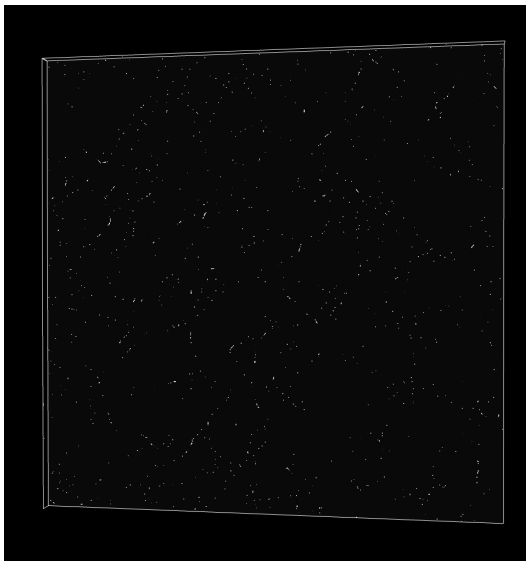
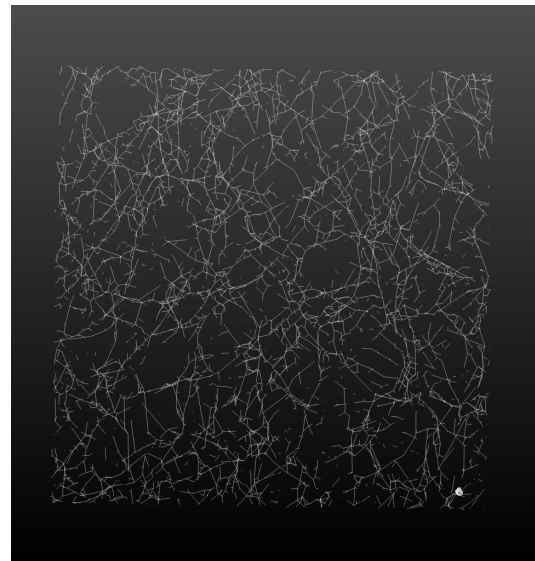
(B) Denoised.



(C) Segmentation.



(D) Distance Map.

(E) Skeletonization ($p = 2$).

(F) Graph.

FIGURE 4.19: Potassium carrageenan spatial graph extraction steps.

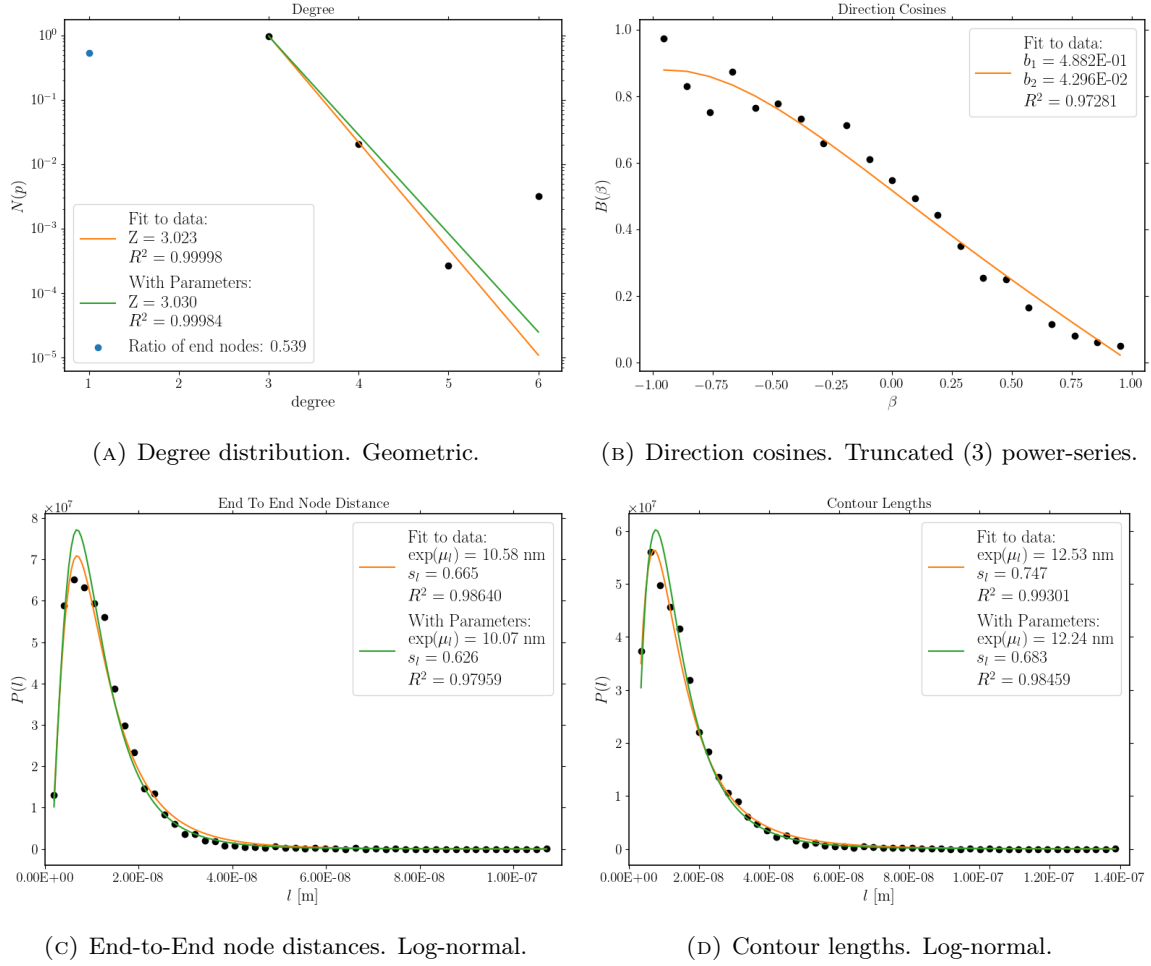


FIGURE 4.20: Statistical distributions (PDF) of Potassium Carrageenan network. Computed properties of the graph are represented in histograms with bins and normalized by area to obtain the PDF. Orange line shows the function with parameters obtained from a non-linear least squares fit to the data. Green lines show the same function with fixed parameters: 4.20a: mean degree (Z) of the network, 4.20c: mean (μ_l) and standard deviation (s_l) of logarithmic end-to-end distances, 4.20d mean (μ_l) and standard deviation (s_l) of logarithmic contour lengths. Histogram bins: end-to-end distances (50), direction cosines (21) contour lengths (50)

4.9.4 Transmission Electron Microscopy: Pectin

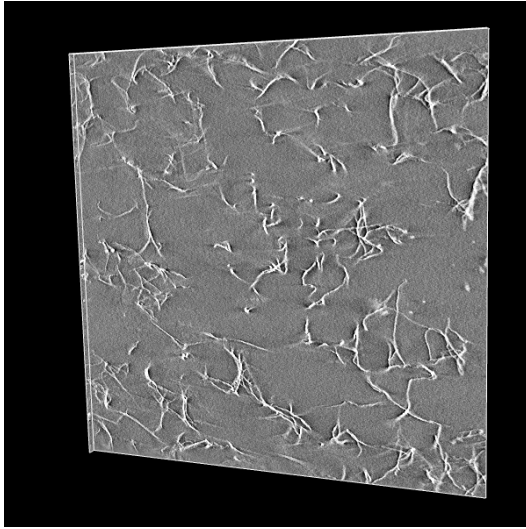
The gelation method exploited herein has been described in detail previously [69] and used in [111]. To summarize: a commercially available pectin sample that had a high degree of methylesterification (DM) (78%) was modified using a processive pectin methyl esterase (PME) extracted from oranges and sourced from Sigma, to obtain a block-wise sample with DM of around 40%, and the resulting polymer freeze dried. The polymeric fine structure was measured using capillary electrophoresis as described elsewhere [69]. Subsequently a solution was made by dissolving 0.02 g of the freeze-dried pectin in 1.5 ml of deionized water.

Next 0.16 g of glucono delta-lactone (GDL) was dissolved in water and mixed with the pectin solution as quickly as possible, in order to minimize any hydrolysis of GDL occurring before mixing, while being careful not to form bubbles. This solution was then loaded into the requisite sample cells for [TEM](#).

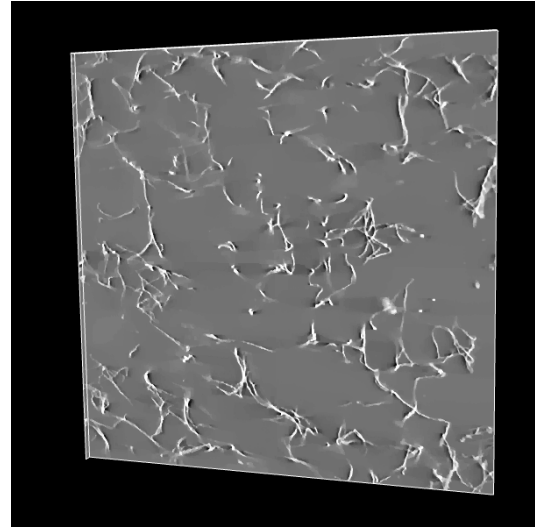
The volumetric tomography image used in this study has a size of $1000 \times 1000 \times 32$ pixels, with resolution $1.72 \times 1.72 \times 1.72$ nm per pixel.

The pipeline is the same than for carragenan [subsection 4.9.3](#), with the exception of GRAPH_SPACING

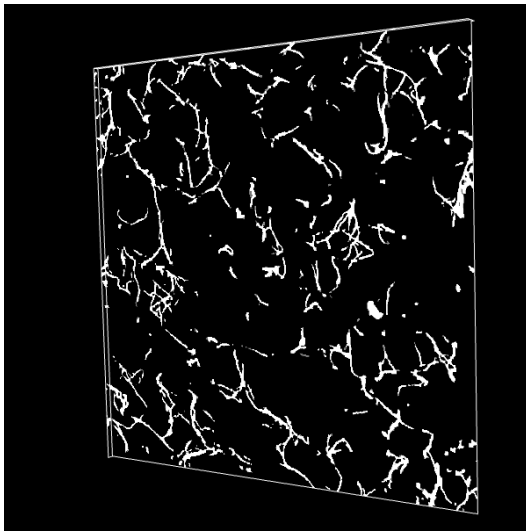
```
# TV denoising
export TV_LAMBDA=20
## Region Growing Binarization
export BIN_RG_LOWER=0.0
export BIN_RG_UPPER=5000
export BIN_SAFE_PERCENTAGE=0.05
## Hole Filling
export HOLE_MAJORITY=3
export HOLE_RADIUS=1
export HOLE_ITERATIONS=10000
## Skeletonization
export SKEL_SELECT=dmax
export SKEL_TYPE=1isthmus
export SKEL_PERSISTENCE=2
## Spatial Graph Extraction
export GRAPH_IGNORE_SHORT_EDGES=2
export GRAPH_SPACING="1.72E-09 1.72E-09 1.72E-09"
## Fit graph data to distributions
export HISTO_ETE_BINS=30
export HISTO_COSINES_BINS=21
export HISTO_CONTOUR_BINS=30
```

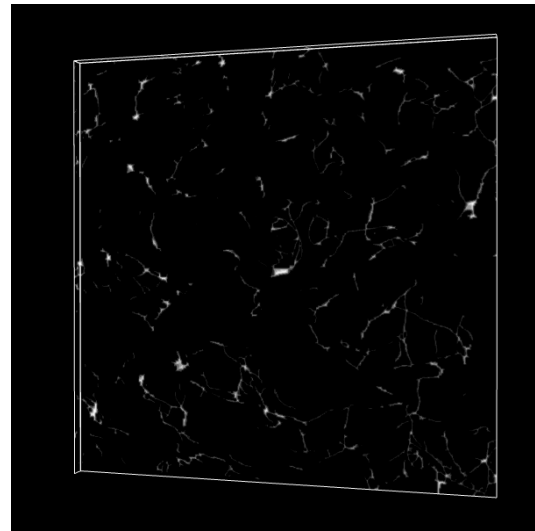
(A) Original volume.



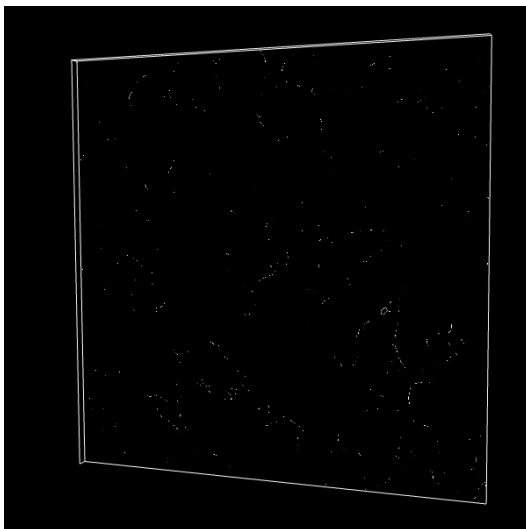
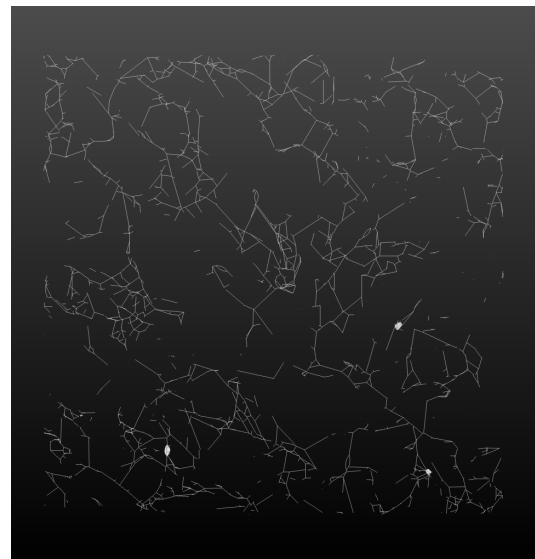
(B) Denoised.



(C) Segmentation.



(D) Distance Map.

(E) Skeletonization ($p = 2$).

(F) Graph.

FIGURE 4.21: Pectin spatial graph extraction steps.

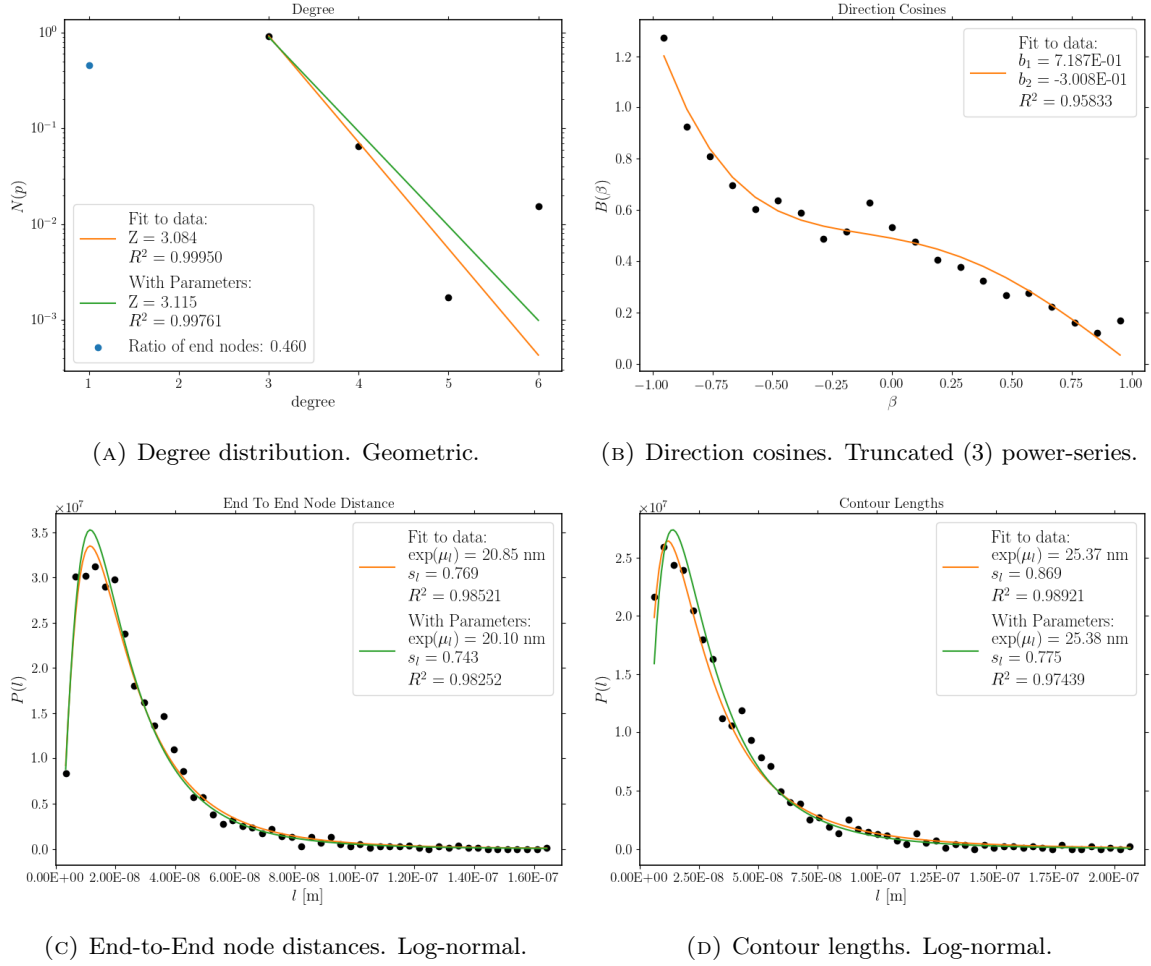


FIGURE 4.22: Statistical distributions (PDF) of Pectin network. Computed properties of the graph are represented in histograms with bins and normalized by area to obtain the PDF. Orange line shows the function with parameters obtained from a non-linear least squares fit to the data. Green lines show the same function with fixed parameters: 4.22a: mean degree (Z) of the network, 4.22c: mean (μ_l) and standard deviation (s_l) of logarithmic end-to-end distances, 4.22d mean (μ_l) and standard deviation (s_l) of logarithmic contour lengths. Histogram bins: end-to-end distances (50), direction cosines (21) contour lengths (50)

4.10 Conclusions

An image pipeline was generated to extract the network architecture of biopolymer networks of different scales, both for proteins and polysaccharides. The network architecture is represented as a **Spatial Graph**, containing the connectivity and the geometric information to fully describe the network geometry.

Computing some graph properties, degree, end-to-end distances, contour lengths and direction cosines of angles of the different biopolymers, it is observed they share the same statistical

distribution for those properties. These distributions were reported already for collagen and fibrin networks [61], but never for polysaccharides. I think this universality in distributions, which suggests independence from the scale of the polymer, originates from the network formation with cross-links.

The spatial graph can be used by theoreticians as a starting scaffold for dynamic simulations, instead of the over-simplified methods currently used in literature (see [section 1.5](#) for a survey). This might help for models able to predict the whole range of dynamics in non-affine networks.

The set of parameters can be divided in two, those related with the segmentation, and those related with the network extraction. The segmentation set includes the denoising and the binarization algorithms. The segmentation should be supervised and non-automatic, but it is easy to get a good result. As a suggestion, better to get false negatives –lose biopolymer information–, than false positives –treat noise as biopolymer–. In the other set of network extraction parameters, the more influential is the persistence parameter –similar to pruning spurious branches–, and it is dependant on the data, it can even be kept at zero for images with low noise in the surface of the biopolymer, but usually values from 1 to 3 are enough in most situations to remove hairy, non-existing branches introduced by noise.

All the code is freely available and open to be re-used, modified and improved by interested users.

5. Reconstructing networks from statistical distributions

5.1 Motivation

This chapter explores the ability to reconstruct, or regenerate an in-silico network solely based on the statistical distributions of a few graph properties extracted from the spatial graph computed from 3D images in [chapter 4](#). The generated 3D networks from the statistical distributions allow the theoretical and computational exploration on how the networks might behave when making fundamental changes on its properties, like length between junctions, orientation, or number of neighbors in the junctions. The proposed implementation is flexible, allowing to take into account any computable property from a spatial graph.

5.2 Methods

The software used in this chapter, has been independently developed here, but is based on the idea of Lindström et al. [\[94\]](#) about reconstructing the network architecture of polymers networks from a statistical description. They were based on the work of Yeong and Torquato [\[113, 114\]](#) about reconstructing random porous media. This work generalizes the implementation of Lindström et al. [\[94\]](#), allowing to add extra statistical distributions associated to any graph property.

The input for the in-silico reconstruction of biopolymer networks is the statistical distributions of 3 properties in this case. This is the minimal set of information required to reconstruct the architecture of the network to study the mechanical properties (but other parameters could be added to study other or more complex situations). For example, if the structure of the

edges / chains is heterogeneous (e.g. chains with different properties, such persistence length), extra statistical distributions can be added with minimal effort.

So in order to regenerate the network in-silico, the following distributions should be obtained from the image analysis:

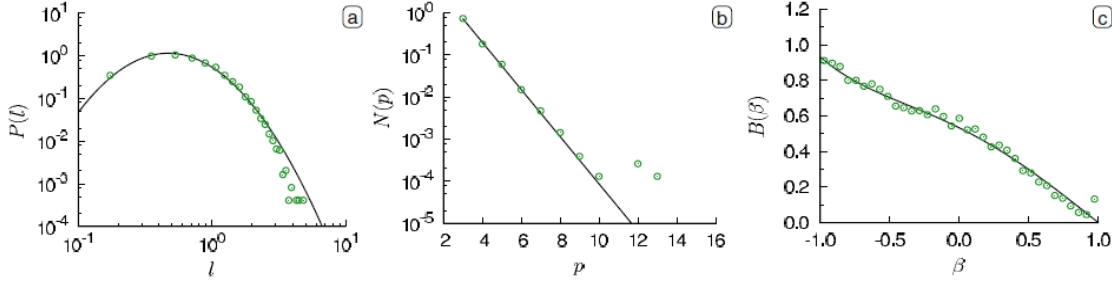


FIGURE 5.1: Statistical distributions for Collagen I proposed by Lindstrom [94]. (a): End-to-end distance between pair of nodes, (b): Degree of nodes, (c): Direction cosine.

1. **Degree of nodes** $N(p)$. i.e. number of edges per node or cross-link.
2. **Length of edges** $P(\ell)$. Length of the edges between connected nodes. There are two options to define this length.
 - End to end vector distance (EtE-length). Consider the edge as a straight line between nodes.
 - Contour length (curved). Take into account all the path of the chain, usually curved.

In fact for many examples the difference between these is less than one might have expected at a first glance. Nisslert et al. [115], who use a different reconstructing algorithm, report that straight and curved lengths have similar distribution, although the straight lines were about a 15% shorter.

I expect that high flexibility of chains and/or low density of cross-links would increase the differences between them. Biopolymers tend to be semi-flexible, stiff enough to keep this difference low. The difference between these lengths can be used as a measure of the persistence length (ℓ_p) of the chains.

3. **Direction cosine** $B(\beta)$ between incident edges. Takes into account the relative orientation of the edges incident on the same node.

As described in this thesis, these distributions can be obtained using **image analysis** of data from [confocal microscopy](#), or transmission electron microscopy ([TEM](#)). See chapter 4 for more information about this.

The distributions used herein to develop the reconstruction software are those seen originally in Lindstrom's paper [94]. These distributions were selected because they fitted the data taken from image analysis of collagen I, but no claim about their use in other systems was made.

It is worth reiterating that with the work described in this thesis, experimentally obtained distributions describing the architectures of biopolymer networks can be extracted. Whatever functional forms are found to faithfully represent the experimental data can be used to reconstruct in-silico networks. However, given the similarities in the distributions found in [chapter 4](#) for very different biopolymer networks the following distributions are chosen to use in the reconstruction:

1. $N(p)$. The **node degree** modelled with a shifted geometric distribution.

$$N(p) = q(1 - q)^{p-3} \quad (5.1)$$

where $q = 1/(Z - 2)$, p is the degree of the node - how many other nodes it is directly connected to - and Z is the average node degree.

The distribution is shifted because the minimum degree of a node is $p = 3$. The case $p = 1$ corresponds to dead-ends nodes, and $p = 2$ correspond to bending nodes. Dead-ends can be ignored because they don't affect to the mechanical properties of the bulk network. And the bending nodes can be ignored if their 2 edges are merged into a unique, larger edge, connecting the neighbors of this bending node between them. The contour length is lost in this process if the edges are only defined by end to end vectors.

2. $P(\ell)$ The **length distribution** modelled as a logarithmic-normal distribution.

$$P(\ell) = \frac{1}{\ell s \sqrt{2\pi}} \exp \left\{ \left[- \frac{(\mu - \ln \ell)^2}{2s^2} \right] \right\} \quad (5.2)$$

where μ and s are the mean and standard deviation of $\ln \ell$ respectively. A gamma distribution $P(\ell) = \frac{\ell^{\alpha-1} \exp(-\ell/\beta)}{\Gamma(\alpha)\beta^\alpha}$ is also feasible to describe the lengths. The log-normal is usually preferred because it fits better in most cases and its two parameters relate directly with computable network parameters.

3. $B(\beta)$ The **direction cosines distribution** has been modelled by a truncated power series.

$$B(\beta) = \sum_{k=1}^m b_k (1 - \beta)^{2k-1} \quad (5.3)$$

Truncating it at $m = 3$, and with $B(\beta)$ normalized to unity, then there are two parameters left, b_1 and b_2 (or other combination).

These 3 distributions are enough to approximate the architecture of a network of collagen if the main interest is in its mechanical properties. So, the network can be reconstructed with five independent parameters: μ , s , Z , b_1 and b_2 .

The output of the algorithm constructed is a network or spatial [Graph](#), which is formed by 2 sets: a set of nodes, containing the nodes position and the degree; and a second set of edges, which connect pair of nodes. The software does not allow parallel edges, i.e. two edges connected the same pair of nodes, and the network is fully connected, there are no nodes or clusters of nodes isolated from the rest of the network.

5.3 Euclidean Graph Generation Algorithm

The Euclidean graph generation (EGG) algorithm reconstructs a 3D network on a periodic cube domain Σ of size Λ . The algorithm uses a Monte Carlo method with Simulated annealing, this is a general heuristic method for global optimization which avoids the convergence to local minimums on the energy landscape.

The following steps describe the algorithm:

- I** An **initial graph configuration** H_0 is generated by placing nodes drawn from a randomly uniform distribution inside the domain Σ . Then a valency drawn from the degree distribution $N(p)$ is assigned to each node. And finally, pairs of nodes are connected by edges, but not exceeding the fixed degree of any node. The initial configuration does not follow the length, or the direction cosine distribution.

In practice, the complex systems library Igraph is used to generate H_0 with the described constraints, see the attached code documentation for more details.

- II** There are two possible movements to **update** the graph.

- a) Remove at random two edges from H and then add other two edges without changing the valence of the nodes. This hard constraint only allows 2 possible edge configurations.
- b) Move the position of a node a random distance in the range $[0, \rho]$, where ρ can be tuned to enhance convergence.

Note that the valence of the nodes is unchanged by any of these updates.

III A non-negative “**energy**” **function** in a graph is defined as: $E(H) = A_P(H) + A_B(H)$, where A_P and A_B are the Cramer-von Mises test statistics for the distributions P and B respectively. The Cramer-von Mises test [116] compares a set of variables $x_1 < x_2 < \dots < x_n$ with a distribution f and produces an associated value A_f . The smaller A_f the better the set of data $\{x\}$ fit the distribution f .

With this definition of energy, $E(H)$ has a global minimum for graphs with the target length, $P(\ell)$ and direction cosine $B(\beta)$ distributions.

IV To find the graph with minimal $E(H)$, a **simulated annealing** algorithm is used. Starting at $H = H_0$, it attempts to update the graph to H' with the equally probable transitions **a)** or **b)**. If the energy of the new graph is lower than before $E(H') \leq E(H)$ the transition is accepted. If the energy of the new graph is greater, $E(H') > E(H)$, the transition still can be accepted with a probability $\exp\{[E(H) - E(H')]/T\}$. Here T is analogous to temperature, and it is the main characteristic of simulated annealing algorithms. This temperature decays exponentially with the number of accepted transitions, which reduces progressively the number of “unfavorable” accepted transitions, but avoids the network getting stuck in local energy minima.

V **Convergence** criteria. The algorithm will stop when the energy $E(H)$ is close to zero or smaller than a threshold. The threshold will depend on the number of nodes (size) of the network.

5.4 Results

A code engine was written and has been successfully tested, generating a network from the three statistical distributions described. To test it, I have used the distributions that Lindström et al. [94] propose for their the collagen data. In Fig.5.2 a reduction of 99.9% from

the initial graph energy $E(H_0)$ to the final graph is observed. Fig. 5.3 shows how the final state successfully follows the target distributions. Note that the simulated and target-distributions are superimposed with each other, no fit is involved.

The resulting in-silico reconstructed network is just one instance of the many configurations that follow the same statistical distributions. For validation of the results against real in-vivo networks, in Lindström et al. [61], they use finite element methods (FEM) to compute simple and differential shear modules based on the geometry of the reconstructed networks. The results compare favourably with the experimental measurements of shear obtained from in-vivo networks.

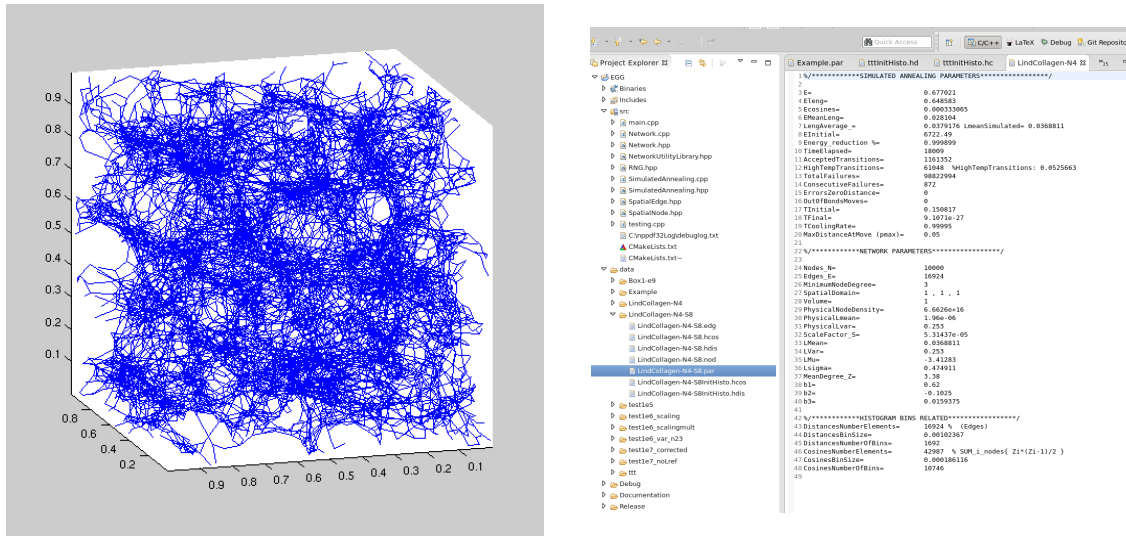


FIGURE 5.2: Reconstructed collagen network Visualization of reconstructed collagen network using Matlab. Simulation parameters from the reconstruction algorithm.

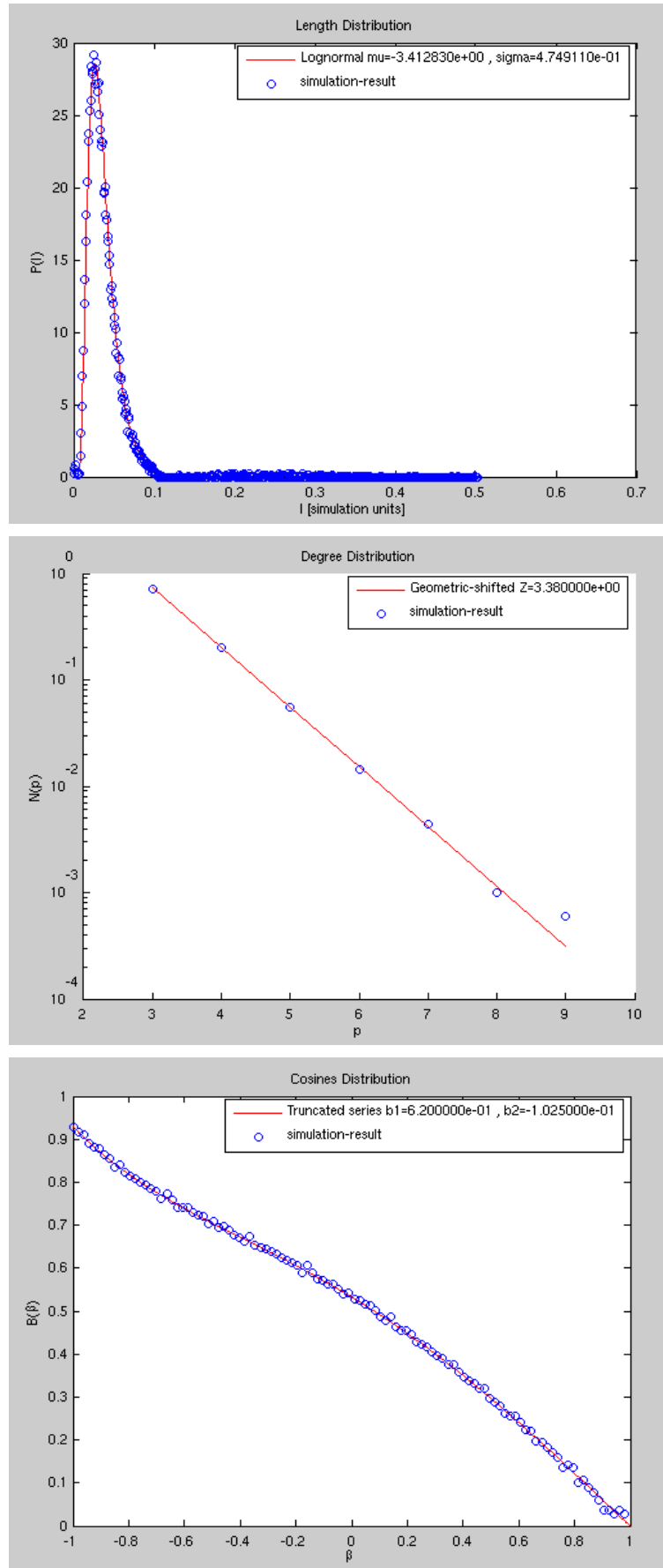


FIGURE 5.3: Comparison between target distributions and simulated results in the reconstructed collagen network. Length Distribution $P(\ell)$. Degree distribution $N(p)$. Direction cosines distribution $B(\beta)$.

6. Conclusions and Future Work

6.1 Scope of Thesis

The importance of the network architecture for the mechanical properties of biopolymer networks is often obscured by focussing on the properties of single-components. [2], demonstrated that the nonlinear response of a strained network can be explained taking into account the non-linear properties of single-biopolymer and assuming that the network architecture is isotropic and homogeneous. This bottom-up approach has proven successful for a wide range of biopolymers [15], but might not be enough to explain the detailed onset of strain-stiffening and the exponent of the non-linear response, as shown recently with other protein networks[117]. An alternative explanation to non-linear elasticity was given with the focus on network architecture in the material[23], pointing out that the fine-grain control of non-linear elasticity depended on single-biopolymer components, **the linkage characteristics**, and **the geometry of the connections**.

Non-affine deformations are at the core of explaining the complex behaviour of biopolymer networks, but methods to reliably gather the geometry of the networks have been lacking and are the focus of this thesis. Affine regimes have been shown to be capable in characterizing the mechanical properties of bulk and dense materials, but in biology this is not always the case.

6.2 Summary

I rely on 3D imaging, tomography or Z stacks of images, to extract the network geometry more reliably than would be possible from 2D images. I provided tools for image analysis in 3D in a high-performance computing language, and contributed them, fully tested and documented to the de-facto standard image analysis library ITK ([chapter 2](#)).

Gathering the network structure from images requires that the information extracted is reliable. In [chapter 3](#) several distinct polysaccharide gels networks were realized. A set of these networks was further treated for study by transmission electron microscopy (TEM), which requires additional preparation. Another set of the same networks, this time unadulterated, was studied by small-angle x-ray scattering (SAXS). The comparison between the two techniques showed that careful protocols can produce datasets where information acquired by imaging above around 20 nm is broadly consistent with that obtained by SAXS studies carried out on unadulterated samples. The fact that at larger length scales the structure of the networks is preserved in the TEM samples, allow us to extract and quantify the network architecture and connectivity with confidence, information that is lost in scattering experiments owing to the intrinsic averaging nature of the technique.

I pursued the task of extracting the network architectures from images in [chapter 4](#) using a pipeline of image analysis to denoise and binarize the original images. I then developed state-of-the-art algorithms taken from the digital topology community to skeletonize the image, being aware of not changing the topology of the original network in the process. With a thin-image and introducing novel tools in this area, I created a one-to-one map between the image and a spatial graph, connecting in this way the image analysis field with the area of complex networks. I am now capable of studying and characterizing our networks of biopolymers with a wider range of tools.

I computed the degree, end-to-end node distances, contour lengths, angle, and direction cosines of those angles to characterize our sample networks of different biopolymers, ranging from actin filaments, to polysaccharides, such as pectin and carrageenan. The statistical distributions of these properties show strong similarities between protein and polysaccharides networks. Such universalities should be further explored to understand their origin.

I also provided a simulated annealing tool in [chapter 5](#) to recreate networks completely in-silico given a set of statistical distributions: degree, end-to-end distance, and director cosines. This allows the theoretical exploration of networks in-silico just changing the input distributions. Future work in this area might include adding distributions for other graph properties for finer control.

6.3 Conclusions

I have focused in bringing computation tools to the community (all the work is open source, tested and documented) where we can start investigating low density, non-affine networks from a more analytical point of view, I foresee theoreticians and modellers taking these networks **extracted from experimental results** as starting points for their models, instead of relying on the more homogeneous mikado model, and providing a greater scale approach than lattice models.

- Validated transmission electron microscopy (TEM) imaging as a reliable tool to examine polysaccharides network geometries after its comparison with small-angle x-ray scattering.
- Provided a mapping between images and spatial graphs, allowing the study of these networks with the tools from the complex systems and networks field.
- Found universalities of certain graph properties in the geometry of gel-like biopolymer networks at different scales.
- Provided a tool to reconstruct networks in-silico based on statistical distribution. This allows the exploration of different network architectures that can be used in modelling and simulating behaviour of these networks under different conditions.

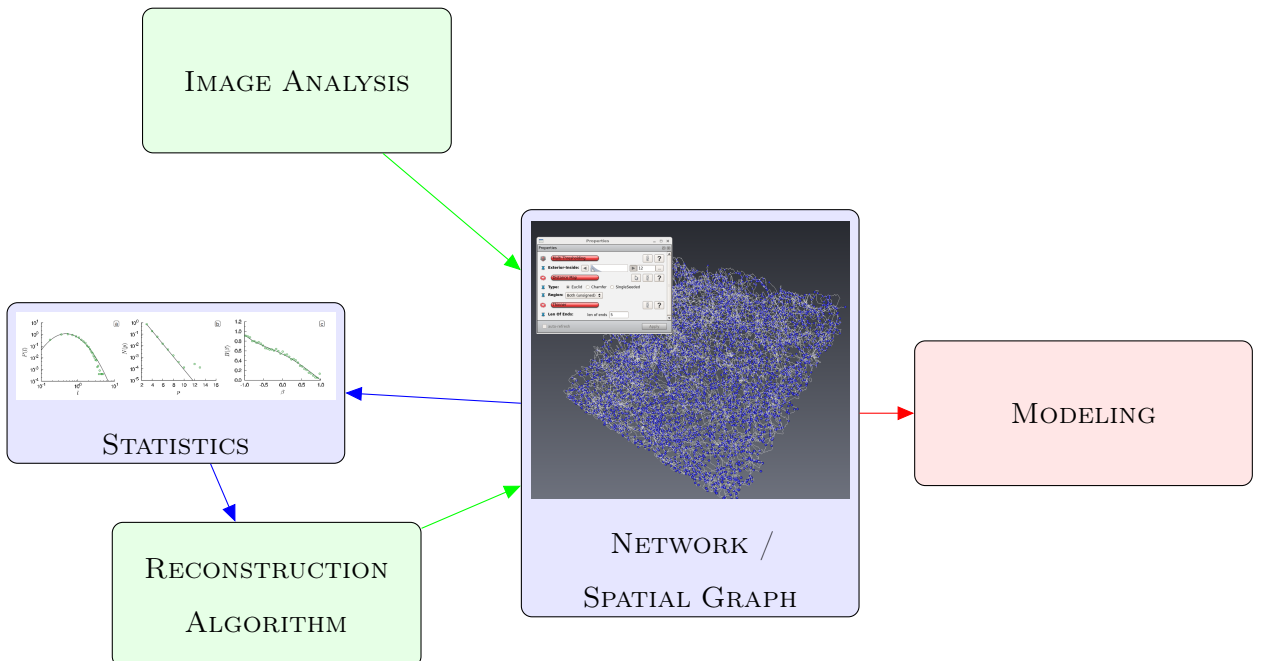


FIGURE 6.1: Scheme of the project.

6.4 Future work

6.4.1 Network Formation

A simulation of network formation might be pursued in the future using a growing spatial graph, a dynamic population of cross-linker and monomers, and a persistence length parameter representing the stiffness of the biopolymer.

Such a simulation could begin with a fixed set of randomly distributed ‘embryonic’ fibers, composed by two end-nodes and an edge between them. At each step of the simulation, a node which represents one end-point of a fiber, would be selected at random from the population, and would grow with a probability given by the current monomer population. If the growth occurs, the monomer population would be reduced –until depletion of the solution. The direction of the growth would depend on the comparison between the persistence length parameter and the position of edge points attached to that node. In this way stiff biopolymer would tend to grow in straight lines, and flexible chains would tend to bend. When two chains are close, and growth occurs, they would attach with a probability depending on the current population of cross-linkers. If the cross-link does not happen, the chain will grow without colliding. Also, at every step, every occupied position, by an edge point or a node, which is close to another occupied position would have a probability to attach depending again on the cross-link population. The network generation would stop when there are no monomer left. Optionally, another parameter would be a detachment probability of cross-linked fibers. Also pseudo-forces can be simulated where edges stores tension every time they attach to another fiber, and release that tension to their adjacent edges when they are released. This might shed some light into slow-dynamical processes and the quake-events behaviours reported [118]. This tension could also be added randomly to existing attached edges to simulate energy transfer from Brownian movements of the components. After stability of the network, its graph properties can be studied, and compare them to the statistical distributions found in our image-extracted networks.

6.4.2 Long time behaviour: network quakes and aging

There are reports in the literature [119], and also by group mates[118, 120], that at long times, physical biopolymer networks can be affected by sudden de-correlations that occur over a few

tenths of a second. This is driven by the dynamic of the junction zones between chains. It could be either that just thermal fluctuations are opening and closing these junctions zones, or that internal stresses are driving them. The latter is reported experimentally in our group with micro-rheology DWS techniques [118], where such quake-like events are associated with the release of chain constraints due to unbinding of cross-links. Current models of networks, such as the Glassy WLC model [121], cannot predict this behaviour [120]. A computer simulation as sketched in [subsection 6.4.1](#) may shed some light on the topic.

It is also interesting the aging phenomena: the slow change of the physical properties over time. This behaviour is related with non-ergodic systems and with glassy systems, where long orders correlations(solids) no longer exists, but there is still some local order [122].

This work provides the tools to study such local order in junction zones, with the full geometry and connectivity provided by the spatial graph.

6.4.3 ITKBoostGraph

The work contributed here about mapping between images and a spatial graph could be integrated in the future in the image analysis library ITK as an external module to facilitate its adoption.

6.4.4 Complex Networks Tools

The spatial graphs can be further analyzed with tools from the complex networks field, computing quantities like clustering and centrality are two lines of code away from the current status using the boost graph library (BGL). The spatial graph can be already exported to the graphviz format to be used by any other network library for the purpose of further analysis.

6.4.5 Public Database

Set-up or contribute to a centralized public database of volumetric images of biopolymer networks, where graphs, analysis, skeletonized images, and extracted spatial graphs would be associated.

A. Publications and Open Source Contributions

A.1 Open Source Contributions

A.1.1 ITKIsotropicWavelets

For Insight ToolKit: ITKIsotropicWavelets External Module <https://github.com/phcerdan/ITKIsotropicWavelets/>

This contribution to ITK provides a multi-resolution framework based on a steerable wavelet pyramid [44] with all the isotropic mother wavelets available in the literature [46, 49]. It also couple the pyramid with the Riesz function [47] for directional analysis. Specially useful for the study of textures [123] and phase analysis [46].

In this thesis, this module work has been used to aid the segmentation of the networks as a pre-processing step before binarization. It improves the signal to noise ratio, helping to binarize the image to capture the network.

I have contributed with more than 20.000 lines of c++ code and documentation. This work has been integrated into ITK as an external module since ITK version 4.13, this allow the user to build the module directly from the ITK. Also wrappings for the c++ code to use it from python are available.

A.1.2 Radial Intensity FFT

The code used in [chapter 2](#), including a graphical interface for the study of the radial intensity of a Fourier Transform is available in <https://github.com/phcerdan/FFTRadialIntenstiy>.

The author has contributed to around 3000 lines of c++ code and 1000 python code, including a notebook that can be used to reproduce exactly the work-flow and figures shown in [chapter 3](#), find the notebook here:

https://github.com/phcerdan/FFTRadialIntenstiy/tree/master/resultsPaper/figures_jupyternb

A.1.3 FFT Radial Intensity

Software used in [chapter 3](#) to create a radial intensity plot from a Fourier Transformed image. It gives the average intensity of all the pixels of an FFT image at a certain frequency. If the FFT image is shifted, i.e. with the zero frequency pixel at the center of the image, it is equivalent to averaging the value of the pixels that intersect the perimeter of concentric rings of different radius, with the radius representing the modulo of the frequency. Useful to compare images with scattering data. It can be used as console script or with a graphical interface.

<https://github.com/phcerdan/FFTRadialIntenstiy>

A.1.4 DGtal: Critical Kernels Framework

Pull request with review of the contribution: <https://github.com/DGtal-team/DGtal/pull/1147>

The contribution to DGtal implements the skeletonization (or thinning) recently proposed in the literature [[93](#), [103](#)] based on critical kernels frameworks, to perform thinning conserving the topology of the original object.

Ready to use tools using this framework can be found in: <https://github.com/DGtal-team/DGtalTools>

The work was reviewed, accepted and merged in the library since version 0.9.4.

I have contributed with more than 7000 lines of c++ code, documentation and scripts for applications.

A.1.5 Spatial Graph Extraction

The code for the spatial graph extraction is available with a Mozilla Public License (open source) in https://github.com/phcerdan/object_to_spatial_graph.

This includes scrips in c++ and python to perform the image pipeline, the skeletonization script using DGtal (see [subsection A.1.4](#)), converting the raw graph to a spatial graph, computing graph properties and statistical distributions, as well as the code for generating the figures in [section 4.9](#). It uses ITK, VTK, Boost, DGtal and proxTV libraries.

Python scripts for plotting and fitting statistical distributions (see [Figure 4.18](#)) can be found in the mentioned repository. Concretely here:

https://github.com/phcerdan/object_to_spatial_graph/tree/master/src/python-scripts

I have contributed with around 8000 lines of c++ and python code.

A.1.6 Open Science

All the work done in this thesis is openly available, and can be reproduced and modified by anyone. I have tried to split my work into chunks and contribute them into existing libraries to maximize its re-usability.

I also provide notebooks with the analysis, data and plots created in [chapter 3](#).

The notebook of [chapter 4](#) is not provided due to the analysis was done between two computers, with the thin/skeletonization process was carried in a workstation with more memory. But the chapter should provide enough information about the pipeline and the details to reproduce it.

Users can contact me in any of these public github repositories with any inquiry or issue, and can contribute adding new features.

Glossary

L_c Contour length. The maximum end-to-end distance of a linear polymer chain. For a single-strand polymer molecule, this usually means the end-to-end distance of the chain extended to the all-trans conformation. For chains with complex structure, only an approximate value of the contour length may be accessible.. [1](#), [113](#)

L_e Deflection or entanglement length. In the tube model picture: typical distance between two collision points of a “test-polymer” with its surrounding chains[\[124\]](#). If one approximates the effect of the surrounding medium by a cylindrical tube of diameter d (of the order of magnitude of the mesh size) the entanglement length is given by Odijk’s estimate [\[125\]](#) $L_e \simeq d^2 \ell_p$. [113](#)

ℓ_p Persistence length. Distance over which the angular correlation decreases to $1/e$ of its initial value. Equivalently, typical length scale for the decay of tangent-tangent correlations: $\langle \mathbf{t}(s) \cdot \mathbf{t}(s') \rangle = \exp[-(s - s')/\ell_p]$, where s is the arc-length of the chain, and $\mathbf{t}(s) = \partial \mathbf{r}(s)/\partial s$ is the tangent vector.
 $\ell_p = 2\kappa_{\perp}/((d - 1)k_B T)$ in d -dimensional space where κ_{\perp} is the bending stiffness of the chain [\[124\]](#). [1](#), [100](#), [113](#)

ξ_m Mesh-size. [113](#)

AFM Atomic force microscopy. High resolution type of scanning probe microscopy. It has a lateral resolution of $< 1\text{nm}$ and height resolution of $< 1\text{\AA}$. García and Pérez [\[126\]](#) contains a very interesting review using AFM in polysaccharides. A more general study about dynamical modes can be found in Rief et al. [\[127\]](#). [xi](#), [4](#), [113](#)

BFS Breadth First Search. Strategy to visit branches of a tree or a graph. When facing a branching point, it chooses a branch and follows it reaches another branching point, then it returns to the last branch and follows the rest of the non-visited branches. When

all the branches of the first round have been visited, it continues with the next batch.
The other common strategy is the Depth First Search.. [113](#)

confocal microscopy Confocal microscopy. [101](#), [113](#)

DFS Depth First Search. Strategy to visit branches of a tree or a graph. When facing a branching point, it chooses a branch and follows it in depth, even until it finds another branch point. When it reaches a dead end, it returns to the last branch point and follows another branch. The other common strategy is the Breadth First Search.. [73](#), [113](#)

DFT Discrete Fourier Transform. Transformation of a finite sequence of inputs into the frequency domain. [113](#)

FFT Fast Fourier Transform. Algorithm computing the discrete Fourier transform DFT, reducing the complexity from $\mathcal{O}(n^2)$ of a naive DFT implementation to $\mathcal{O}(n \log n)$. [113](#)

Graph Mathematical graph, set of nodes connected between them by edges.. [60](#), [72](#), [102](#), [113](#)

SAXS Small angle x-ray scattering. [113](#)

SEM Scanning electron microscopy. [xi](#), [113](#)

Spatial Edge Structure associated to edges in a Spatial Graph. It contains an ordered set of points, reflecting the index or position from the image.. [73](#), [113](#)

Spatial Graph Graph with spatial information. Providing position of nodes, and the edge point positions for edges. plural. [viii](#), [ix](#), [58–60](#), [73](#), [74](#), [97](#), [113](#)

Spatial Node Structure associated to vertices in a Spatial Graph. It contains the index or position of the node in the image.. [73](#), [113](#)

TEM Tomography electron microscopy. [xi](#), [84](#), [90](#), [91](#), [95](#), [101](#), [113](#)

References

- [1] Pierre-Gilles de Gennes. “"Pierre-Gilles de Gennes - Nobel Lecture: Soft Matter". Nobelprize.Org. Nobel Media AB 2013.” URL: http://www.nobelprize.org/nobel_prizes/physics/laureates/1991/gennes-lecture.html.
- [2] Cornelis Storm, Jennifer J. Pastore, F. C. MacKintosh, T. C. Lubensky, and Paul A. Janmey. “Nonlinear Elasticity in Biological Gels.” en. In: *Nature* 435.7039 (May 2005). 00641, pp. 191–194. DOI: [10.1038/nature03521](https://doi.org/10.1038/nature03521).
- [3] Michael Rubinstein. *Polymer Physics*. 01859. Oxford ; New York, 2003.
- [4] Erich Schuster. “Hierarchical Structure Function Models of Biopolymer Networks: Thesis Submitted to the Institute of Fundamental Sciences, Massey University, New Zealand, in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Physics, Palmerston North, October 2011.” 00000. PhD thesis. Massey University, 2011. URL: <http://muir.massey.ac.nz/handle/10179/2908>.
- [5] Hsiao-Ping Hsu, Wolfgang Paul, and Kurt Binder. “Breakdown of the Kratky-Porod Wormlike Chain Model for Semiflexible Polymers in Two Dimensions.” en. In: *EPL (Europhysics Letters)* 95.6 (Sept. 2011). 00027, p. 68004. DOI: [10.1209/0295-5075/95/68004](https://doi.org/10.1209/0295-5075/95/68004).
- [6] John F. Marko and Eric D. Siggia. “Stretching Dna.” In: *Macromolecules* 28.26 (1995). 01662, pp. 8759–8770. URL: <http://pubs.acs.org/doi/abs/10.1021/ma00130a008>.
- [7] K Svoboda and S M Block. “Biological Applications of Optical Forces.” In: *Annual Review of Biophysics and Biomolecular Structure* 23.1 (1994). 01672, pp. 247–285. DOI: [10.1146/annurev.bb.23.060194.001335](https://doi.org/10.1146/annurev.bb.23.060194.001335).
- [8] Karel Svoboda, Christoph F. Schmidt, Bruce J. Schnapp, and Steven M. Block. “Direct Observation of Kinesin Stepping by Optical Trapping Interferometry.” en. In: *Nature* 365.6448 (Oct. 1993), pp. 721–727. DOI: [10.1038/365721a0](https://doi.org/10.1038/365721a0).

- [9] Janshoff, Neitzert, Oberdörfer, and Fuchs. “Force Spectroscopy of Molecular Systems-Single Molecule Spectroscopy of Polymers and Biomolecules.” ENG. In: *Angewandte Chemie (International ed. in English)* 39.18 (Sept. 2000), pp. 3212–3237.
- [10] Piotr E. Marszalek, Yuan-Ping Pang, Hongbin Li, Jamal El Yazal, Andres F. Oberhauser, and Julio M. Fernandez. “Atomic Levers Control Pyranose Ring Conformations.” en. In: *Proceedings of the National Academy of Sciences* 96.14 (June 1999). 00149 PMID: 10393918, pp. 7894–7898. DOI: [10.1073/pnas.96.14.7894](https://doi.org/10.1073/pnas.96.14.7894).
- [11] M D Wang, H Yin, R Landick, J Gelles, and S M Block. “Stretching DNA with Optical Tweezers.” In: *Biophysical Journal* 72.3 (Mar. 1997), pp. 1335–1346. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1184516/>.
- [12] R G Haverkamp, A T Marshall, and M A K Williams. “Model for Stretching Elastic Biopolymers Which Exhibit Conformational Transformations.” eng. In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 75.2 Pt 1 (Feb. 2007), p. 021907.
- [13] Pierre Gilles de Gennes. *Scaling Concepts in Polymer Physics*. 11981. Ithaca, N.Y, 1979.
- [14] Christopher W. Macosko. *Rheology: Principles, Measurements, and Applications*. Advances in interfacial engineering series. 02872. New York, NY: VCH, 1994.
- [15] Jan-Michael Y. Carrillo, Fred C. MacKintosh, and Andrey V. Dobrynin. “Nonlinear Elasticity: From Single Chain to Networks and Gels.” In: *Macromolecules* 46.9 (May 2013). 00003, pp. 3679–3692. DOI: [10.1021/ma400478f](https://doi.org/10.1021/ma400478f).
- [16] F. C. MacKintosh, J. Käs, and P. A. Janmey. “Elasticity of Semiflexible Biopolymer Networks.” In: *Physical Review Letters* 75.24 (Dec. 1995), pp. 4425–4428. DOI: [10.1103/PhysRevLett.75.4425](https://doi.org/10.1103/PhysRevLett.75.4425).
- [17] M. Sheinman, C. P. Broedersz, and F. C. MacKintosh. “Nonlinear Effective-Medium Theory of Disordered Spring Networks.” In: *Physical Review E* 85.2 (Feb. 2012). 00000. DOI: [10.1103/PhysRevE.85.021801](https://doi.org/10.1103/PhysRevE.85.021801).
- [18] Norman Y. Yao, Daniel J. Becker, Chase P. Broedersz, Martin Depken, Frederick C. MacKintosh, Martin R. Pollak, and David A. Weitz. “Nonlinear Viscoelasticity of Actin Transiently Cross-Linked with Mutant Alpha-Actinin-4.” In: *Journal of Molecular Biology* 411.5 (Sept. 2011). 00000, pp. 1062–1071. DOI: [10.1016/j.jmb.2011.06.049](https://doi.org/10.1016/j.jmb.2011.06.049).

- [19] Kyu Hyun, Manfred Wilhelm, Christopher O. Klein, Kwang Soo Cho, Jung Gun Nam, Kyung Hyun Ahn, Seung Jong Lee, Randy H. Ewoldt, and Gareth H. McKinley. “A Review of Nonlinear Oscillatory Shear Tests: Analysis and Application of Large Amplitude Oscillatory Shear (LAOS).” In: *Progress in Polymer Science* 36.12 (Dec. 2011). 00189, pp. 1697–1753. DOI: [10.1016/j.progpolymsci.2011.02.002](https://doi.org/10.1016/j.progpolymsci.2011.02.002).
- [20] D. Head, A. Levine, and F. MacKintosh. “Distinct Regimes of Elastic Response and Deformation Modes of Cross-Linked Cytoskeletal and Semiflexible Polymer Networks.” In: *Physical Review E* 68.6 (Dec. 2003). 00237. DOI: [10.1103/PhysRevE.68.061907](https://doi.org/10.1103/PhysRevE.68.061907).
- [21] David A. Head, Alex J. Levine, and F. C. MacKintosh. “Deformation of Cross-Linked Semiflexible Polymer Networks.” In: *Physical Review Letters* 91.10 (Sept. 2003), p. 108102. DOI: [10.1103/PhysRevLett.91.108102](https://doi.org/10.1103/PhysRevLett.91.108102).
- [22] E. M. Huisman, T. van Dillen, P. R. Onck, and E. Van der Giessen. “Three-Dimensional Cross-Linked F-Actin Networks: Relation between Network Architecture and Mechanical Behavior.” In: *Physical Review Letters* 99.20 (Nov. 2007). 00000, p. 208103. DOI: [10.1103/PhysRevLett.99.208103](https://doi.org/10.1103/PhysRevLett.99.208103).
- [23] P R Onck, T Koeman, T van Dillen, and E van der Giessen. “Alternative Explanation of Stiffening in Cross-Linked Semiflexible Networks.” eng. In: *Physical Review Letters* 95.17 (Oct. 2005), p. 178102.
- [24] Jan Wilhelm and Erwin Frey. “Elasticity of Stiff Polymer Networks.” In: *Physical Review Letters* 91.10 (Sept. 2003). 00000, p. 108103. DOI: [10.1103/PhysRevLett.91.108103](https://doi.org/10.1103/PhysRevLett.91.108103).
- [25] Qi Wen, Anindita Basu, Paul A. Janmey, and Arjun G. Yodh. “Non-Affine Deformations in Polymer Hydrogels.” en. In: *Soft Matter* 8.31 (2012). 00018, p. 8039. DOI: [10.1039/c2sm25364j](https://doi.org/10.1039/c2sm25364j).
- [26] Anindita Basu, Qi Wen, Xiaoming Mao, T. C. Lubensky, Paul A. Janmey, and A. G. Yodh. “Nonaffine Displacements in Flexible Polymer Networks.” In: *Macromolecules* 44.6 (Mar. 2011). 00025, pp. 1671–1679. DOI: [10.1021/ma1026803](https://doi.org/10.1021/ma1026803).
- [27] L. G. Rizzi, S. Auer, and D. A. Head. “Importance of Non-Affine Viscoelastic Response in Disordered Fibre Networks.” en. In: *Soft Matter* 12.19 (2016). 00000, pp. 4332–4338. DOI: [10.1039/C6SM00139D](https://doi.org/10.1039/C6SM00139D).
- [28] D. Head, A. Levine, and F. MacKintosh. “Mechanical Response of Semiflexible Networks to Localized Perturbations.” In: *Physical Review E* 72.6 (Dec. 2005). DOI: [10.1103/PhysRevE.72.061914](https://doi.org/10.1103/PhysRevE.72.061914).

- [29] Preethi L Chandran and Victor H Barocas. “Affine versus Non-Affine Fibril Kinematics in Collagen Networks: Theoretical Studies of Network Behavior.” eng. In: *Journal of biomechanical engineering* 128.2 (Apr. 2006), pp. 259–270. DOI: [10.1115/1.2165699](https://doi.org/10.1115/1.2165699).
- [30] Jeffrey S. Palmer and Mary C. Boyce. “Constitutive Modeling of the Stress–strain Behavior of F-Actin Filament Networks.” In: *Acta Biomaterialia* 4.3 (May 2008). 00074, pp. 597–612. DOI: [10.1016/j.actbio.2007.12.007](https://doi.org/10.1016/j.actbio.2007.12.007).
- [31] Jan Åström, Sami Saarinen, Kaarlo Niskanen, and Juhani Kurkijärvi. “Microscopic Mechanics of Fiber Networks.” In: *Journal of Applied Physics* 75.5 (1994), p. 2383. DOI: [10.1063/1.356259](https://doi.org/10.1063/1.356259).
- [32] Erich Schuster, Leif Lundin, and Martin AK Williams. “Investigating the Relationship between Network Mechanics and Single-Chain Extension Using Biomimetic Polysaccharide Gels.” In: *Macromolecules* 45.11 (2012), pp. 4863–4869. URL: <http://pubs.acs.org/doi/abs/10.1021/ma300724n>.
- [33] Enrico Conti and Fred MacKintosh. “Cross-Linked Networks of Stiff Filaments Exhibit Negative Normal Stress.” In: *Physical Review Letters* 102.8 (Feb. 2009). DOI: [10.1103/PhysRevLett.102.088102](https://doi.org/10.1103/PhysRevLett.102.088102).
- [34] A. Sharma, A. J. Licup, K. A. Jansen, R. Rens, M. Sheinman, G. H. Koenderink, and F. C. MacKintosh. “Strain-Controlled Criticality Governs the Nonlinear Mechanics of Fibre Networks.” en. In: *Nature Physics* 12.6 (June 2016), pp. 584–587. DOI: [10.1038/nphys3628](https://doi.org/10.1038/nphys3628).
- [35] A. Sharma, M. Sheinman, K. M. Heidemann, and F. C. MacKintosh. “Elastic Response of Filamentous Networks with Compliant Crosslinks.” In: *arXiv* (2013). URL: <http://arxiv.org/abs/1309.3876>.
- [36] Jan A. Åström, P. B. Sunil Kumar, Ilpo Vattulainen, and Mikko Karttunen. “Strain Hardening, Avalanches, and Strain Softening in Dense Cross-Linked Actin Networks.” en. In: *Physical Review E* 77.5 (May 2008). 00040. DOI: [10.1103/PhysRevE.77.051913](https://doi.org/10.1103/PhysRevE.77.051913).
- [37] C. P. Broedersz, M. Sheinman, and F. C. MacKintosh. “Filament-Length-Controlled Elasticity in 3D Fiber Networks.” In: *Physical Review Letters* 108.7 (Feb. 2012). 00040. DOI: [10.1103/PhysRevLett.108.078102](https://doi.org/10.1103/PhysRevLett.108.078102).
- [38] Pierre Ronceray, Chase Broedersz, and Martin Lenz. “Fiber Networks Amplify Active Stress.” In: *arXiv preprint arXiv:1507.05873* (2015). 00000. URL: <http://arxiv.org/abs/1507.05873>.

- [39] Mahsa Vahabi, Abhinav Sharma, Albert James Licup, Anne S. G. van Oosten, Peter A. Galie, Paul A. Janmey, and Fred C. MacKintosh. “Elasticity of Fibrous Networks under Uniaxial Prestress.” en. In: *Soft Matter* 12.22 (2016), pp. 5050–5060. DOI: [10.1039/C6SM00606J](https://doi.org/10.1039/C6SM00606J).
- [40] *Cubic Crystal System*. en. Mar. 2018. URL: https://en.wikipedia.org/w/index.php?title=Cubic_crystal_system&oldid=828866627.
- [41] Fanlong Meng and Eugene M. Terentjev. “Nonlinear Elasticity of Semiflexible Filament Networks.” en. In: *Soft Matter* 12.32 (Aug. 2016). 00003, pp. 6749–6756. DOI: [10.1039/C6SM01029F](https://doi.org/10.1039/C6SM01029F).
- [42] Paul A. Janmey, Margaret E. McCormick, Sebastian Rammensee, Jennifer L. Leight, Penelope C. Georges, and Fred C. MacKintosh. “Negative Normal Stress in Semiflexible Biopolymer Gels.” In: *Nature Materials* 6.1 (Dec. 2006), pp. 48–51. DOI: [10.1038/nmat1810](https://doi.org/10.1038/nmat1810).
- [43] W. T. Freeman and E. H. Adelson. “The Design and Use of Steerable Filters.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.9 (Sept. 1991). 03462, pp. 891–906. DOI: [10.1109/34.93808](https://doi.org/10.1109/34.93808).
- [44] Eero P. Simoncelli and William T. Freeman. “The Steerable Pyramid: A Flexible Architecture for Multi-Scale Derivative Computation.” In: *ICIP (3)*. 00964. 1995, pp. 444–447. URL: <http://www.merl.com/papers/docs/TR95-15.pdf>.
- [45] Peter Kovesi. “Image Features from Phase Congruency.” In: *Videre: Journal of computer vision research* 1.3 (1999). 01142, pp. 1–26. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.5658&rep=rep1&type=pdf>.
- [46] S. Held, M. Storath, P. Massopust, and B. Forster. “Steerable Wavelet Frames Based on the Riesz Transform.” In: *IEEE Transactions on Image Processing* 19.3 (Mar. 2010). 00062, pp. 653–667. DOI: [10.1109/TIP.2009.2036713](https://doi.org/10.1109/TIP.2009.2036713).
- [47] N. Chenouard and M. Unser. “3D Steerable Wavelets in Practice.” In: *IEEE Transactions on Image Processing* 21.11 (Nov. 2012). 00016, pp. 4522–4533. DOI: [10.1109/TIP.2012.2206044](https://doi.org/10.1109/TIP.2012.2206044).
- [48] M. Unser, N. Chenouard, and D. Van De Ville. “Steerable Pyramids and Tight Wavelet Frames in $L_2(\mathcal{R}^d)$.” In: *IEEE Transactions on Image Processing* 20.10 (Oct. 2011). 00000, pp. 2705–2721. DOI: [10.1109/TIP.2011.2138147](https://doi.org/10.1109/TIP.2011.2138147).

- [49] Pedram Pad, Virginie Uhlmann, and Michael Unser. “VOW: Variance-Optimal Wavelets for the Steerable Pyramid.” In: *constraints* 3 (2014). 00007, p. 4. URL: <http://bigwww.epfl.ch/publications/pad1401.pdf>.
- [50] Stephane G. Mallat. “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.” In: *IEEE transactions on pattern analysis and machine intelligence* 11.7 (1989). 21999, pp. 674–693. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=192463.
- [51] Christopher E. Heil and David F. Walnut. “Continuous and Discrete Wavelet Transforms.” In: *SIAM review* 31.4 (1989). 01229, pp. 628–666. URL: <http://epubs.siam.org/doi/abs/10.1137/1031129>.
- [52] Tao Qian, Mang I. Vai, and Yuesheng Xu. *Wavelet Analysis and Applications*. en. Springer Science & Business Media, Feb. 2007.
- [53] Akram Aldroubi, Carlos Cabrelli, and Ursula M. Molter. “Wavelets on Irregular Grids with Arbitrary Dilation Matrices, and Frames Atoms for $L^2(R^d)$.” In: *arXiv:math/0703438* (Mar. 2007). arXiv: [math / 0703438](http://arxiv.org/abs/math/0703438). URL: <http://arxiv.org/abs/math/0703438>.
- [54] Javier Portilla and Eero P. Simoncelli. “Image Denoising via Adjustment of Wavelet Coefficient Magnitude Correlation.” In: *Image Processing, 2000. Proceedings. 2000 International Conference On*. Vol. 3. 00044. IEEE, 2000, pp. 277–280. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=899349.
- [55] M. Unser, D. Sage, and D. Van De Ville. “Multiresolution Monogenic Signal Analysis Using the Riesz-Laplace Wavelet Transform.” In: *IEEE Transactions on Image Processing* 18.11 (Nov. 2009). 00000, pp. 2402–2418. DOI: [10.1109/TIP.2009.2027628](https://doi.org/10.1109/TIP.2009.2027628).
- [56] Michael Felsberg and Gerald Sommer. “The Monogenic Signal.” In: *IEEE Transactions on Signal Processing* 49.12 (2001). 00593, pp. 3136–3144. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=969520.
- [57] M. Unser and D. Van De Ville. “Wavelet Steerability and the Higher-Order Riesz Transform.” In: *IEEE Transactions on Image Processing* 19.3 (Mar. 2010). 00051, pp. 636–652. DOI: [10.1109/TIP.2009.2038832](https://doi.org/10.1109/TIP.2009.2038832).

- [58] Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. “Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain.” In: *IEEE Transactions on Image processing* 12.11 (2003). 02184, pp. 1338–1351. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1240101.
- [59] T.. Blu and F.. Luisier. “The SURE-LET Approach to Image Denoising.” In: *IEEE Transactions on Image Processing* 16.11 (Nov. 2007). 00272, pp. 2778–2786. DOI: [10.1109/TIP.2007.906002](https://doi.org/10.1109/TIP.2007.906002).
- [60] Zsuzsanna Puspoki and Michael Unser. “Template-Free Wavelet-Based Detection of Local Symmetries.” In: *IEEE Transactions on Image Processing* 24.10 (Oct. 2015). 00004, pp. 3009–3018. DOI: [10.1109/TIP.2015.2436343](https://doi.org/10.1109/TIP.2015.2436343).
- [61] Stefan B. Lindström, Artem Kulachenko, Louise M. Jawerth, and David A. Vader. “Finite-Strain, Finite-Size Mechanics of Rigidly Cross-Linked Biopolymer Networks.” en. In: *Soft Matter* 9.30 (2013), p. 7302. DOI: [10.1039/c3sm50451d](https://doi.org/10.1039/c3sm50451d).
- [62] Niklas Lorén, Liubov Shtykova, Siw Kidman, Patrik Jarvoll, Magnus Nydén, and Anne-Marie Hermansson. “Dendrimer Diffusion in κ -Carrageenan Gel Structures.” In: *Biomacromolecules* 10.2 (Feb. 2009), pp. 275–284. DOI: [10.1021/bm801013x](https://doi.org/10.1021/bm801013x).
- [63] Bernhard Walther, Niklas Lorén, Magnus Nydén, and Anne-Marie Hermansson. “Influence of κ -Carrageenan Gel Structures on the Diffusion of Probe Molecules Determined by Transmission Electron Microscopy and NMR Diffusometry.” In: *Langmuir* 22.19 (Sept. 2006). 00023, pp. 8221–8228. DOI: [10.1021/la061348w](https://doi.org/10.1021/la061348w).
- [64] Davide Magatti, Matteo Molteni, Barbara Cardinali, Mattia Rocco, and Fabio Ferri. “Modeling of Fibrin Gels Based on Confocal Microscopy and Light-Scattering Data.” In: *Biophysical Journal* 104.5 (Mar. 2013), pp. 1151–1159. DOI: [10.1016/j.bpj.2013.01.024](https://doi.org/10.1016/j.bpj.2013.01.024).
- [65] Alice C. Dohnalkova, Matthew J. Marshall, Bruce W. Arey, Kenneth H. Williams, Edgar C. Buck, and James K. Fredrickson. “Imaging Hydrated Microbial Extracellular Polymers: Comparative Analysis by Electron Microscopy.” en. In: *Applied and Environmental Microbiology* 77.4 (Feb. 2011), pp. 1254–1262. DOI: [10.1128/AEM.02001-10](https://doi.org/10.1128/AEM.02001-10).
- [66] Leif Lundin and Anne-Marie Hermansson. “Rheology and Microstructure of Ca- and Na-K-Carrageenan and Locust Bean Gum Gels.” In: *Carbohydrate Polymers* 34.4 (Dec. 1997), pp. 365–375. DOI: [10.1016/S0144-8617\(97\)00119-7](https://doi.org/10.1016/S0144-8617(97)00119-7).

- [67] Francesco Brun, Agostino Accardo, Maurizio Marchini, Fulvia Ortolani, Gianluca Turco, and Sergio Paoletti. “Texture Analysis of TEM Micrographs of Alginate Gels for Cell Microencapsulation.” eng. In: *Microscopy Research and Technique* 74.1 (Jan. 2011), pp. 58–66. DOI: [10.1002/jemt.20874](https://doi.org/10.1002/jemt.20874).
- [68] David S. Domozych, Iben Sørensen, Zoë A. Popper, Julie Ochs, Amanda Andreas, Jonatan U. Fangel, Anna Pielach, Carly Sacks, Hannah Brechka, Pia Ruisi-Besares, William G. T. Willats, and Jocelyn K. C. Rose. “Pectin Metabolism and Assembly in the Cell Wall of the Charophyte Green Alga *Penium margaritaceum*.” en. In: *Plant Physiology* 165.1 (May 2014), pp. 105–118. DOI: [10.1104/pp.114.236257](https://doi.org/10.1104/pp.114.236257).
- [69] Bradley W. Mansel, Che-Yi Chu, Andrew Leis, Yacine Hemar, Hsin-Lung Chen, Leif Lundin, and Martin A. K. Williams. “Zooming in: Structural Investigations of Rheologically Characterized Hydrogen-Bonded Low-Methoxyl Pectin Networks.” In: *Biomacromolecules* 16.10 (Oct. 2015), pp. 3209–3216. ISSN: 1525-7797. DOI: [10.1021/acs.biomac.5b00870](https://doi.org/10.1021/acs.biomac.5b00870). URL: <https://doi.org/10.1021/acs.biomac.5b00870>.
- [70] Yariv Wine, Noa Cohen-Hadar, Amihay Freeman, and Felix Frolov. “Elucidation of the Mechanism and End Products of Glutaraldehyde Crosslinking Reaction by X-Ray Structure Analysis.” eng. In: *Biotechnology and Bioengineering* 98.3 (Oct. 2007), pp. 711–718. DOI: [10.1002/bit.21459](https://doi.org/10.1002/bit.21459).
- [71] Clarence Sterling. “Crystal-Structure of Ruthenium Red and Stereochemistry of its Pectic Stain.” In: *American Journal of Botany* 57.2 (1970), pp. 172–175. ISSN: 0002-9122. DOI: [10.2307/2440510](https://doi.org/10.2307/2440510). URL: <https://www.jstor.org/stable/2440510>.
- [72] Hajime Tanaka, Takafumi Hayashi, and Toshio Nishi. “Application of Digital Image Analysis to Pattern Formation in Polymer Systems.” In: *Journal of Applied Physics* 59.11 (1986/06//6/1/1986). 00122, p. 3627.
- [73] Atsuo Takahashi, Rio Kita, Takashi Shinozaki, Kenji Kubota, and Makoto Kaibara. “Real Space Observation of Three-Dimensional Network Structure of Hydrated Fibrin Gel.” en. In: *Colloid and Polymer Science* 281.9 (Feb. 2003). 00023, pp. 832–838. DOI: [10.1007/s00396-002-0839-0](https://doi.org/10.1007/s00396-002-0839-0).
- [74] Hans J. Johnson, M. McCormick, L. Ibáñez, and The Insight Software Consortium. *The ITK Software Guide*. Third. 00015 *In press*. Kitware, Inc., 2017.

- [75] Pradeep K. Luther. “Sample Shrinkage and Radiation Damage of Plastic Sections.” en. In: *Electron Tomography*. Springer, New York, NY, 2007, pp. 17–48. DOI: [10.1007/978-0-387-69008-7_2](https://doi.org/10.1007/978-0-387-69008-7_2). URL: https://link.springer.com/chapter/10.1007/978-0-387-69008-7_2.
- [76] Nigel Chaffey. *Principles and Techniques of Electron Microscopy: Biological Applications*. 4th. JSTOR, 2001.
- [77] Alvaro Barbero and Suvrit Sra. “Fast Newton-Type Methods for Total Variation Regularization.” In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 313–320. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Barbero_226.pdf.
- [78] Álvaro Barbero and Suvrit Sra. “Modular Proximal Optimization for Multidimensional Total-Variation Regularization.” In: *arXiv:1411.0589 [math, stat]* (Nov. 2014). arXiv: [1411.0589 \[math, stat\]](https://arxiv.org/abs/1411.0589). URL: <http://arxiv.org/abs/1411.0589>.
- [79] Boshra Rajaei. “An Analysis and Improvement of the BLS-GSM Denoising Method.” en. In: *Image Processing On Line* 4 (Apr. 2014). 00010, pp. 44–70. DOI: [10.5201/ipol.2014.86](https://doi.org/10.5201/ipol.2014.86).
- [80] Pablo Hernandez-Cerdan. “Isotropic and Steerable Wavelets in N Dimensions. A Multiresolution Analysis Framework for ITK.” In: *arXiv:1710.01103 [cs]* (Oct. 2017). arXiv: [1710.01103 \[cs\]](https://arxiv.org/abs/1710.01103). URL: <http://arxiv.org/abs/1710.01103>.
- [81] Ivan Usov and Raffaele Mezzenga. “FiberApp: An Open-Source Software for Tracking and Analyzing Polymers, Filaments, Biomacromolecules, and Fibrous Objects.” In: *Macromolecules* 48.5 (Mar. 2015). 00001, pp. 1269–1280. DOI: [10.1021/ma502264c](https://doi.org/10.1021/ma502264c).
- [82] Mehmet Sezgin and Bulent Sankur. “Survey over Image Thresholding Techniques and Quantitative Performance Evaluation.” In: *Journal of Electronic Imaging* 13.1 (2004). 02429, pp. 146–168. DOI: [10.1117/1.1631315](https://doi.org/10.1117/1.1631315).
- [83] S.K Pal, R.A King, and A.A Hashim. “Automatic Grey Level Thresholding through Index of Fuzziness and Entropy.” In: *Pattern Recognition Letters* 1.3 (Mar. 1983). 00202, pp. 141–146. DOI: [10.1016/0167-8655\(83\)90053-3](https://doi.org/10.1016/0167-8655(83)90053-3).
- [84] Sankar K. Pal and Azriel Rosenfeld. “Image Enhancement and Thresholding by Optimization of Fuzzy Compactness.” In: *Pattern Recognition Letters* 7.2 (1988), pp. 77–86. URL: <http://www.sciencedirect.com/science/article/pii/0167865588901225>.

- [85] Xiang Bai, Longin Jan Latecki, and Wen-Yu Liu. “Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution.” In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29.3 (2007), pp. 449–462. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4069261.
- [86] Polina Golland, W. Eric, and L. Grimson. “Fixed Topology Skeletons.” In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference On*. Vol. 1. 2000, pp. 10–17. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=855792.
- [87] Yaorong Ge and J.M. Fitzpatrick. “On the Generation of Skeletons from Discrete Euclidean Distance Maps.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.11 (1996). 00158, pp. 1055–1066. DOI: [10.1109/34.544075](https://doi.org/10.1109/34.544075).
- [88] R.L. Ogniewicz and O. Kübler. “Hierarchic Voronoi Skeletons.” In: *Pattern Recognition* 28.3 (Mar. 1995), pp. 343–359. DOI: [10.1016/0031-3203\(94\)00105-U](https://doi.org/10.1016/0031-3203(94)00105-U).
- [89] Kálmán Palágyi, Emese Balogh, Attila Kuba, Csongor Halmai, Balázs Erd\Hohelyi, Erich Sorantin, and Klaus Hasegger. “A Sequential 3D Thinning Algorithm and Its Medical Applications.” In: *Information Processing in Medical Imaging*. 00152. Springer, 2001, pp. 409–415. URL: http://link.springer.com/chapter/10.1007/3-540-45729-1_42.
- [90] Kálmán Palágyi, Gábor Németh, and Péter Kardos. “Topology Preserving Parallel 3D Thinning Algorithms.” In: *Digital Geometry Algorithms*. Ed. by Valentin E. Brimkov and Reneta P. Barneva. Vol. 2. 00000. Dordrecht: Springer Netherlands, 2012, pp. 165–188. URL: http://www.springerlink.com/index/10.1007/978-94-007-4174-4_6.
- [91] Michel Couprie and Gilles Bertrand. “Asymmetric Parallel 3D Thinning Scheme and Algorithms Based on Isthmuses.” In: *Pattern Recognition Letters*. Special Issue on Skeletonization and its Application 76 (June 2016), pp. 22–31. DOI: [10.1016/j.patrec.2015.03.014](https://doi.org/10.1016/j.patrec.2015.03.014).
- [92] David Coeurjolly, Bertrand Kerautret, Jacques-Olivier Lachaud, troussil, Pierre Gueth, rolanddenis, Jérémy Levallois, isivigno, Martial Tola, Kacper Pluta, PHCerdan, cgurps, aline martin, xprov, Guillaume Damian, Adrien Krähenbühl, Jérémy Gaillard, Nicolas Normand, Monir Hadji, Check your git settings! AUBRY Nicolas, Boris Mansencal, Jean-David Gènevaux, Marc Chevalier, vanthonguyen, and Bitdeli Chef. *DGtal-Team/DGtal*:

- Release 0.9.4.1*. Apr. 2018. DOI: [10.5281/zenodo.1211065](https://doi.org/10.5281/zenodo.1211065). URL: <https://zenodo.org/record/1211065>.
- [93] Gilles Bertrand and Michel Couprie. *Parallel Skeletonization Algorithms in the Cubic Grid Based on Critical Kernels*. Elsevier, 2017.
- [94] Stefan B. Lindström, David A. Vader, Artem Kulachenko, and David A. Weitz. “Biopolymer Network Geometries: Characterization, Regeneration, and Elastic Properties.” en. In: *Physical Review E* 82.5 (Nov. 2010), p. 051905. DOI: [10.1103/PhysRevE.82.051905](https://doi.org/10.1103/PhysRevE.82.051905).
- [95] Michael Mitzenmacher. “A Brief History of Generative Models for Power Law and Lognormal Distributions.” en. In: *Internet Mathematics* 1.2 (Jan. 2004), pp. 226–251. DOI: [10.1080/15427951.2004.10129088](https://doi.org/10.1080/15427951.2004.10129088).
- [96] Steven A. Frank. “The Common Patterns of Nature.” In: *Journal of evolutionary biology* 22.8 (Aug. 2009), pp. 1563–1585. DOI: [10.1111/j.1420-9101.2009.01775.x](https://doi.org/10.1111/j.1420-9101.2009.01775.x).
- [97] Steven Frank. “How to Read Probability Distributions as Statements about Process.” en. In: *Entropy* 16.11 (Nov. 2014), pp. 6059–6098. DOI: [10.3390/e16116059](https://doi.org/10.3390/e16116059).
- [98] Andrew M. Stein, David A. Vader, Louise M. Jawerth, David A. Weitz, and Leonard M. Sander. “An Algorithm for Extracting the Network Geometry of Three-Dimensional Collagen Gels.” In: *Journal of Microscopy* 232.3 (2008), pp. 463–475. URL: <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2818.2008.02141.x/full>.
- [99] Gilles Bertrand and Michel Couprie. “Powerful Parallel and Symmetric 3D Thinning Schemes Based on Critical Kernels.” en. In: *Journal of Mathematical Imaging and Vision* 48.1 (Jan. 2014). 00009, pp. 134–148. DOI: [10.1007/s10851-012-0402-7](https://doi.org/10.1007/s10851-012-0402-7).
- [100] Andrew M. Stein. “Mathematical Models for Glioma Spheroid Invasion in Three-Dimensional Collagen-I Gel.” PhD thesis. ProQuest, 2007.
- [101] J. Wu, B. Rajwa, D. L. Filmer, C. M. Hoffmann, B. Yuan, C. Chiang, J. Sturgis, and J. P. Robinson. “Automated Quantification and Reconstruction of Collagen Matrix from 3D Confocal Datasets.” In: *Journal of microscopy* 210.2 (2003), pp. 158–165. URL: <http://onlinelibrary.wiley.com/doi/10.1046/j.1365-2818.2003.01191.x/full>.

- [102] Céline Fouard, E. Cassot, G. Malandain, C. Mazel, S. Prohaska, D. Asselot, M. Westerhoff, and Jean-Pierre Marc-Vergnes. “Skeletonization by Blocks for Large 3D Datasets: Application to Brain Microcirculation.” In: *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium On*. 00023. IEEE, 2004, pp. 89–92. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1398481.
- [103] Michel Couprie and Gilles Bertrand. “A 3D Sequential Thinning Scheme Based on Critical Kernels.” en. In: *Mathematical Morphology and Its Applications to Signal and Image Processing*. Ed. by Jón Atli Benediktsson, Jocelyn Chanussot, Laurent Najman, and Hugues Talbot. Lecture Notes in Computer Science no. 9082. 00000. Springer International Publishing, May 2015, pp. 549–560. DOI: [10.1007/978-3-319-18720-4_46](https://doi.org/10.1007/978-3-319-18720-4_46). URL: http://link.springer.com/chapter/10.1007/978-3-319-18720-4_46.
- [104] PHCerdan. *rieszWaveletPhaseAnalysis*. Mar. 2018. URL: <https://github.com/phcerdan/ITKfilters/blob/3369e1a12f782a9f235fcaa5f6c7b9f956b7f742/scripts-cpp/rieszWaveletPhaseAnalysis.cxx>.
- [105] PHCerdan. *Denoise_input*. Mar. 2018. URL: https://github.com/phcerdan/ITKfilters/blob/3369e1a12f782a9f235fcaa5f6c7b9f956b7f742/scripts-python/denoise_input.py.
- [106] PHCerdan. *regionGrowingSegmentation*. Mar. 2018. URL: <https://github.com/phcerdan/ITKfilters/blob/3369e1a12f782a9f235fcaa5f6c7b9f956b7f742/scripts-cpp/regionGrowingSegmentation.cxx>.
- [107] PHCerdan. *Binary_denoise_3d_fillholes_iterative*. Mar. 2018. URL: https://github.com/phcerdan/ITKfilters/blob/3369e1a12f782a9f235fcaa5f6c7b9f956b7f742/scripts-python/binary_denoise_3d_fillholes_iterative.py.
- [108] PHCerdan. *Thin*. Mar. 2018. URL: https://github.com/phcerdan/object_to_spatial_graph/blob/5a55310d9d9cbd92c1ef128a3fe2e9969d8d2c54/src/thin.cpp.
- [109] PHCerdan. *Analyze_graph*. Mar. 2018. URL: https://github.com/phcerdan/object_to_spatial_graph/blob/5a55310d9d9cbd92c1ef128a3fe2e9969d8d2c54/src/analyze_graph.cpp.
- [110] PHCerdan. *Fit_to_distribution_from_data*. Mar. 2018. URL: https://github.com/phcerdan/object_to_spatial_graph/blob/5a55310d9d9cbd92c1ef128a3fe2e9969d8d2c54/src/python-scripts/fit_to_distribution_from_data.py.

- [111] Pablo Hernandez-Cerdan, Bradley W. Mansel, Andrew Leis, Leif Lundin, and Martin A.K. Williams. “Structural Analysis of Polysaccharide Networks by Transmission Electron Microscopy: Comparison with Small-Angle X-Ray Scattering.” In: *Biomacromolecules* (Jan. 2018). DOI: [10.1021/acs.biomac.7b01773](https://doi.org/10.1021/acs.biomac.7b01773).
- [112] PHCerdan. *Denoise_tv*. Mar. 2018. URL: https://github.com/phcerdan/ITKfilters/blob/3369e1a12f782a9f235fcaa5f6c7b9f956b7f742/scripts-python/denoise_tv.py.
- [113] C. L. Y. Yeong and S. Torquato. “Reconstructing Random Media.” In: *Physical Review E* 57.1 (1998). 00444, p. 495. URL: http://pre.aps.org/abstract/PRE/v57/i1/p495_1.
- [114] C. L. Y. Yeong and S. Torquato. “Reconstructing Random Media. II. Three-Dimensional Media from Two-Dimensional Cuts.” In: *Physical Review E* 58.1 (1998). 00340, p. 224. URL: http://pre.aps.org/abstract/PRE/v58/i1/p224_1.
- [115] Rasmus Nisslert, Mats Kvarnström, Niklas Lorén, Magnus Nydén, and Mats Rudemo. “Identification of the Three-Dimensional Gel Microstructure from Transmission Electron Micrographs.” In: *Journal of Microscopy* 225.1 (2007), pp. 10–21. URL: <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2818.2007.01711.x/full>.
- [116] T. W. Anderson. “On the Distribution of the Two-Sample Cramer-von Mises Criterion.” EN. In: *The Annals of Mathematical Statistics* 33.3 (1962). 00277 Mathematical Reviews number (MathSciNet): MR145607; Zentralblatt MATH identifier: 0116.37601, pp. 1148–1159. DOI: [10.1214/aoms/1177704477](https://doi.org/10.1214/aoms/1177704477).
- [117] Albert James Licup, Stefan Münster, Abhinav Sharma, Michael Sheinman, Louise M. Jawerth, Ben Fabry, David A. Weitz, and Fred C. MacKintosh. “Stress Controls the Mechanics of Collagen Networks.” In: *arXiv preprint arXiv:1503.00924* (2015). 00000. URL: <http://arxiv.org/abs/1503.00924>.
- [118] Bradley W. Mansel and Martin A. K. Williams. “Internal stress drives slow glassy dynamics and quake-like behaviour in ionotropic pectin gels.” en. In: *Soft Matter* 11.35 (Aug. 2015), pp. 7016–7023. ISSN: 1744-6848. DOI: [10.1039/C5SM01720C](https://doi.org/10.1039/C5SM01720C). URL: <https://pubs.rsc.org/en/content/articlelanding/2015/sm/c5sm01720c>.
- [119] Tadashi Kajiyama, Tetsuharu Narita, Véronique Schmitt, François Lequeux, and Laurence Talini. “Slow Dynamics and Intermittent Quakes in Soft Glassy Systems.” en. In: *Soft Matter* 9.46 (Nov. 2013), pp. 11129–11135. DOI: [10.1039/C3SM51992A](https://doi.org/10.1039/C3SM51992A).

- [120] R. R. R. Vincent, B. W. Mansel, A. Kramer, K. Kroy, and M. a. K. Williams. “Micro-Rheological Behaviour and Nonlinear Rheology of Networks Assembled from Polysaccharides from the Plant Cell Wall.” en. In: *New Journal of Physics* 15.3 (Mar. 2013), p. 035002. DOI: [10.1088/1367-2630/15/3/035002](https://doi.org/10.1088/1367-2630/15/3/035002).
- [121] Klaus Kroy and Jens Glaser. “The Glassy Wormlike Chain.” In: *New Journal of Physics* 9.11 (2007), p. 416. URL: <http://iopscience.iop.org/1367-2630/9/11/416>.
- [122] Luca Cipelletti and Laurence Ramos. “Slow Dynamics in Glassy Soft Matter.” In: *arXiv* (), p. 42. URL: <http://arxiv.org/pdf/cond-mat/0502024.pdf>.
- [123] Adrien Depeursinge, Zsuzsanna Puspoki, John Paul Ward, and Michael Unser. “Steerable Wavelet Machines (SWM): Learning Moving Frames for Texture Classification.” eng. In: *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society* (Jan. 2017). 00000. DOI: [10.1109/TIP.2017.2655438](https://doi.org/10.1109/TIP.2017.2655438).
- [124] Erwin Frey, Klaus Kroy, and Jan Wilhelm. “Viscoelasticity Of Biopolymer Networks And Statistical Mechanics Of Semiflexible Polymers.” In: 1998. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.17.5215>.
- [125] Theo Odijk. “The Statistics and Dynamics of Confined or Entangled Stiff Polymers.” In: *Macromolecules* 16.8 (Aug. 1983), pp. 1340–1344. DOI: [10.1021/ma00242a015](https://doi.org/10.1021/ma00242a015).
- [126] Ricardo García and Rubén Pérez. “Dynamic Atomic Force Microscopy Methods.” In: *Surface Science Reports* 47.6–8 (Sept. 2002). 00000, pp. 197–301. DOI: [10.1016/S0167-5729\(02\)00077-8](https://doi.org/10.1016/S0167-5729(02)00077-8).
- [127] Matthias Rief, Filipp Oesterhelt, Berthold Heymann, and Hermann E. Gaub. “Single Molecule Force Spectroscopy on Polysaccharides by Atomic Force Microscopy.” en. In: *Science* 275.5304 (Feb. 1997), pp. 1295–1297. DOI: [10.1126/science.275.5304.1295](https://doi.org/10.1126/science.275.5304.1295).